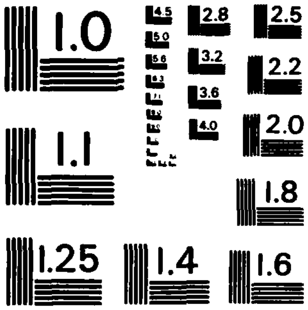
AD-A132 474 COMPUTER-AIDED SYNTHESIS AND DESIGN OF MONOLITHIC MICROWAVE GAAS MESFET AMPLIFIERS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH J T DIJAK AUG 83 1/2

UNCLASSIFIED AFIT/CI/NR-83-19D F/G 9/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Computer-Aided Synthesis and Design of Monolithic Microwave GaAs MESFET Amplifiers

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jerome Thomas Dijak

August 1983

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 83-19D	2. GOVT ACCESSION NO. AD-A132474	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer-Aided Synthesis and Design of Monolithic Microwave GaAs MESFET Amplifiers		5. TYPE OF REPORT & PERIOD COVERED ~~THESIS~~/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerome Thomas Dijak		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Cornell		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		12. REPORT DATE Aug 83
		13. NUMBER OF PAGES 167
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17 1 SEP 1983	Approved for public release: IAW AFR 190-17. Lynn Wolaver LYNN E. WOLAVER Dean for Research and Professional Development Air Force Institute of Technology (ATC) Wright-Patterson AFB OH 45433	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) ATTACHED	DTIC COPY INSPECTED 2	

Accession For

NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification

By
Distribution/

Availability Codes

Dist	Avail and/or Special
A	

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

83 09 13 126

S3-19D

Computer-Aided Synthesis and Design of Monolithic Microwave GaAs MESFET Amplifiers

Jerome Thomas Dijak, Ph.D.
Major, U.S. Air Force
Cornell University 1983

The computer-aided procedures for systematic design of lumped and distributed element even-order amplifier matching networks are extended to include odd-order networks. The gain-bandwidth performance of general odd-order networks as well as that of several specific odd-order topologies of special interest are discussed. Practical monolithic input, output, and interstage matching network topologies using both lumped and distributed elements are shown, and the potential for reducing required chip area by using networks of 3rd order in lieu of 4th order is shown for both low-noise and power amplifier examples.

Analytical techniques are presented for computing the maximum realizable value of the denominator polynomial constant of odd-order gain functions in both the lumped and distributed element cases for 3rd order networks of arbitrary ripple, bandwidth, minimum insertion loss, and gain slope.

Design techniques and practical topologies are presented for matching networks composed only of cascaded transmission lines and shunt open-circuited stubs, for circuits where extremely simple monolithic fabrication is required. Design techniques are presented to overcome the limited control over impedance transformation ratio that is inherent with these circuits.

Tables of circuit element values for several example lumped and distributed element topologies over narrow and wide bandwidths and for several gain slope specifications are included, as well as a computer program for single-frequency design of narrow band networks. FORTRAN source code listings for computer implementations of all major algorithms developed are also presented.

167 pages, bibliography.

Biographical Sketch

Jerome T. Dijak was born in Hammond, Indiana on July 14, 1949. He lived in Highland, Indiana until graduation from high school in 1967, and then attended college at the University of Rochester, Rochester, New York where he earned the Bachelor of Science degree with Distinction in Electrical Engineering in 1971. He received the Master of Science degree in Electrical Engineering from the Air Force Institute of Technology, Dayton, Ohio in 1975.

Mr. Dijak is a member of Tau Beta Pi and a student member of IEEE.

Acknowledgements

The author wishes to thank Professor Walter H. Ku for his guidance in the research leading to this dissertation. Helpful discussions with Professor Zhengfan Li were also appreciated.

This research was supported by the Air Force Rome Air Development Center and the Defense Advanced Research Projects Agency as part of the Space Based Radar Monolithic Module Design and Evaluation Study. It was also supported in part by the Joint Services Electronics Program under contract F49620-81-C-0082.

Contents

Contents (Continued)

Contents (Continued)

Contents (Continued)

List of Figures

List of Figures (Continued)

List of Figures (Continued)

Chapter 1

Introduction

While microwave integrated circuits of hybrid construction allow considerable modifications to a design to optimize performance after fabrication, monolithic microwave integrated circuits (MMIC's) make this impossible in most cases. For this reason, accurate design and fabrication of MMIC's is of much greater importance than ever before with other fabrication technologies.

Of central importance in the MMIC design process is the availability of computer programs to assist the circuit designer in active device modelling, synthesis of impedance matching networks with precisely controlled responses for amplifier gain equalization over arbitrary bandwidths, and amplifier performance analysis and optimization by simulation prior to committing a design for fabrication. This work extends the state of the art in several of these areas, and includes original computer programs that simplify and make more convenient some of the the tasks that must be accomplished during the design of monolithic microwave amplifiers.

Chapter 2 extends the exact lumped element matching network synthesis procedures of Petersen [1] (valid only for even-order gain functions) to allow exact synthesis of equiripple lumped element networks of odd order as well. The unique traits of these odd-order networks are examined, and procedures for taking maximum advantage of these during the synthesis process are presented. Examples are used to illustrate the significant advantages in circuit simplicity and reduced physical size now available to the designer through the use of 3rd order networks in many applications where 4th order matching networks were previously required.

Chapter 3 extends the theory for exact synthesis of distributed element matching networks to include odd-order networks. The unique traits of these distributed element odd-order networks are also examined. Listings of the computer subroutines used to implement the new gain function approximation algorithms for both the distributed and lumped element cases are presented in Appendix A.

The derivation of an analytical method for determining the theoretical limits of the new synthesis parameter available with odd-order functions (the denominator polynomial constant) is presented for 3rd order networks in Chapter 4. The procedure is completely general in that it applies to all 3rd order networks -- regardless of bandwidth, ripple, minimum insertion loss, or gain slope specifications. Analytical methods are presented for both the lumped element and distributed element cases. The computer implementation of these techniques is presented in Appendix A.

Chapter 5 presents design techniques for matching networks allowing only cascaded transmission lines and shunt open-circuited stubs as circuit elements. These networks greatly simplify the fabrication of monolithic amplifier circuits, and the advantages and limitations of this restricted topology class of networks are discussed. Design examples for octave and 10% band monolithic FET amplifiers are presented; the measured RF performance of the octave band amplifier is also presented.

Tables of circuit element values for several example lumped element matching network topologies are presented in Appendix B for both 15% and 100% bandwidths as well as several different gain slopes. Appendix C presents similar tables for several practical distributed element network topologies.

Appendix D presents a concise summary of single-frequency matching techniques, extremely simple lumped element circuits that may be used to implement these for input, output, and interstage networks, and a computer implementation of these procedures for convenience.

Chapter 2

Lumped Element Broadband Matching Network Design (Arbitrary Order)

Computer programs have been available for several years for the synthesis of amplifier impedance matching networks with even-order bandpass gain responses [1-2]. A method for synthesizing odd-order low-pass networks has also been reported recently [3].

The odd-order synthesis method discussed by Riddle and Trew in [3] is based upon the use of even-order low-pass gain functions to define network response, and then adding a multiplicative factor involving the complex variable to the resulting $s_{11}(p)$ function so that an additional circuit element can be extracted during synthesis. Although the authors only discussed this technique in the context of low-pass networks, it could also be applied to bandpass gain functions. The odd-order synthesis techniques described in this work are fundamentally different from those of Riddle and Trew.

This chapter presents a technique for synthesizing lumped element bandpass impedance matching networks with arbitrary gain slope from odd-order gain functions. This is accomplished by extending the even-order synthesis procedure so that odd-order functions may be included while maintaining realizability in the gain function. The procedure can produce narrowband or wideband gain responses with equal facility.

A primary concern in the design of Monolithic Microwave Integrated Circuits (MMIC's) is to find impedance matching networks for the active devices that require a minimum number of elements (for minimum wafer area) as well as providing a means of introducing FET bias voltages with the fewest possible additional RF chokes and DC blocking and bypass capacitors.

All low-pass matching networks, since they are composed only of series inductors and shunt capacitors, require the addition of a physically large shunt inductor, one or more bypass capacitors, and a blocking capacitor, before biasing provisions can be included on-chip; these requirements largely eliminate low-pass networks from GaAs monolithic applications. Bandpass networks, however, allow shunt inductors and series capacitors, so it is possible to employ bandpass topologies that only require the addition of a single DC bypass capacitor to allow on-chip biasing. This feature makes bandpass networks the optimum choice for MMIC's.

2.1. Synthesis Algorithm

Lumped element broadband matching networks with bandpass gain responses of arbitrary gain slope can be synthesized by the following procedure:

(1) Specify a circuit topology (which, in turn, fixes network order).

(2) Specify the desired bandwidth, gain slope, minimum insertion loss, and gain ripple of the network.

(3) Form an equiripple approximation [G(x)] to the desired gain response. (Find unknown coefficients of G(x) by computer algorithm.)

(4) Transform G(x) to $G(\omega^2)$, a transducer power gain function, by substituting $\omega^2 = x$.

(5) Compute $|s_{11}(p)|^2$, then factor into $s_{11}(p)$ and $s_{11}(-p)$, where $s_{11}(p)$ includes all the left half-plane poles and any distribution of the zeroes of $|s_{11}(p)|^2$.

(Use $|s_{11}(p)|^2 = 1 - |s_{21}(p)|^2$, where $p = j\omega$, the complex variable.)

(6) Compute $z_{11}(p)$ from $s_{11}(p)$ by using

$$z_{11}(p) = \frac{1 + s_{11}(p)}{1 - s_{11}(p)}$$

(7) Extract network element values by continued fraction expansion of $z_{11}(p)$.

The following sections will discuss in detail the solution to the gain response approximation problem [step (3)], and a text on network synthesis, such as Weinberg [4], may be consulted for more detail on steps (5) through (7).

2.2. Motivation for Odd Order

The techniques developed by Petersen [1] can be used to synthesize even-order bandpass networks (i.e., N=2,4,6...), and these have been implemented in the CADSYN commercial computer program [2]. The simplest even-order network that can be expected to simultaneously absorb an FET's parasitic capacitance while providing a specific impedance transformation is of 4th order.

Figure 2.1(a) illustrates a typical, practical 4th order bandpass network that can be used to exactly absorb the parasitic output capacitance of an FET (C_p) while providing an impedance transformation between the FET output resistance (R_{out}) and the amplifier load as well as adding gain slope compensation to provide flat gain across the passband despite the FET's intrinsic gain roll-off with frequency. After adding one capacitor, the circuit in Figure 2.1(b) can offer essentially the same RF performance as that in Figure 2.1(a), but with on-chip biasing as well.

We would like to eliminate two of the inductors in the network of Figure 2.1, and use a 3rd order network such as that shown in Figure 2.2 to still provide simultaneous exact parasitic absorption, exact impedance transformation, and bias introduction, while requiring the fewest possible circuit elements.

2.3. Even-Order Equiripple Gain Function Synthesis

The function shown in equation (2-1) will be used to approximate a specified ideal gain response.

$$G(x) = \frac{K * x^k}{P_N(x)} \qquad (2\text{-}1)$$

where: x = the frequency squared variable (ω^2)

k = number of high-pass elements in the network
(determined once topology has been specified)

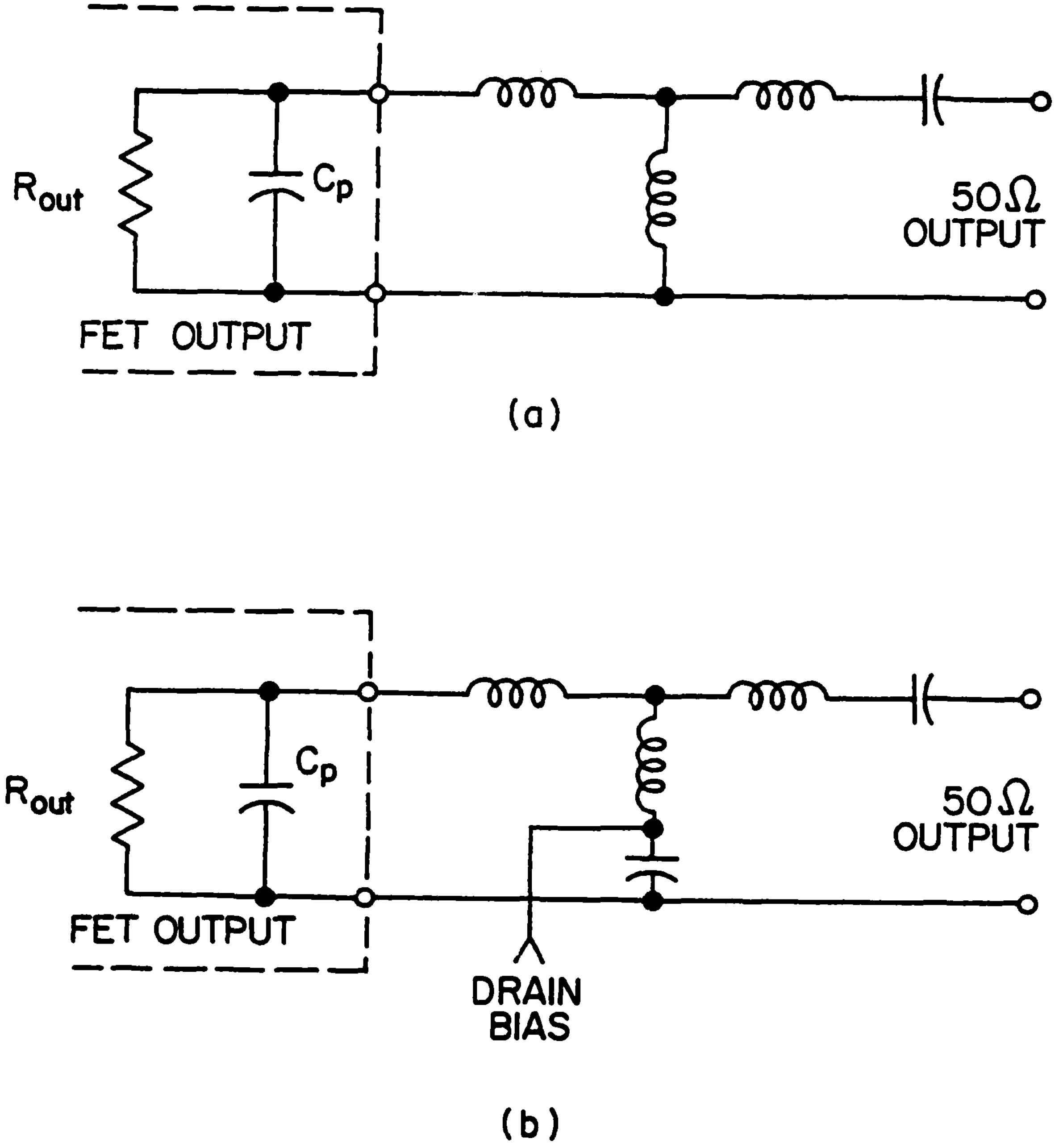


Figure 2.1. 4th Order Bandpass Networks

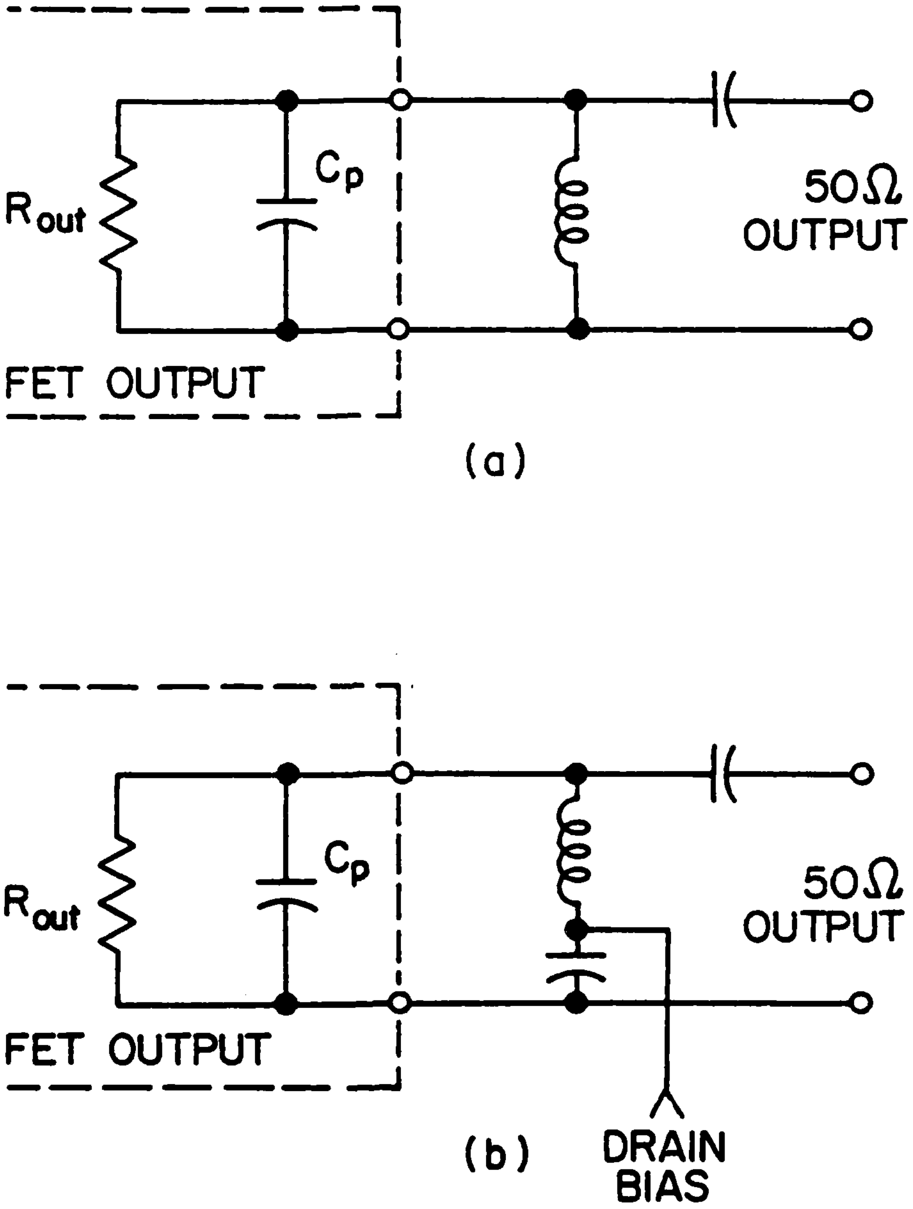


Figure 2.2. 3rd Order Bandpass Networks

K = gain constant chosen to maintain realizability
$(0 \leq G(x) \leq 1 \ \forall \ x > 0)$

$P_N(x)$ = an Nth order polynomial in x, whose coefficients
are to be determined.

An equiripple approximation will be computed by solving for the unknown K and the coefficients of $P_N(x)$.

Our synthesis procedure requires the initial specification of a topology, as well as bandwidth, ripple, gain slope, and minimum insertion loss for the network; the algorithm then proceeds to solve for the coefficients of $P_N(x)$ so that the specified approximation is realized.

The N+1 unknown coefficients of $P_N(x)$ can be determined (for even N) by setting up a system of N+1 equations involving eq (2-1) with the gain value (G(x)) specified at N+1 critical points across the passband alternating above and below the specified ideal gain response, as shown in Figure 2.3 for N = 4. We require G(x) to be decreasing as x increases or decreases away from the passband (so $0 \leq G(x) \leq 1 \ \forall \ x > 0$); this formulation facilitates this behavior in the function for even N, since we can specify an odd number of critical points.

The ideal response shown in Figure 2.3 will be referred to later as the reference gain function, $G_{ref}(x)$, defined as

$$G_{ref}(x) = 10^{\left[\frac{GS}{10} * \frac{\log_{10}(x)}{\log_{10}(2)} \right]}$$

where GS is the specified gain slope in dB per octave.

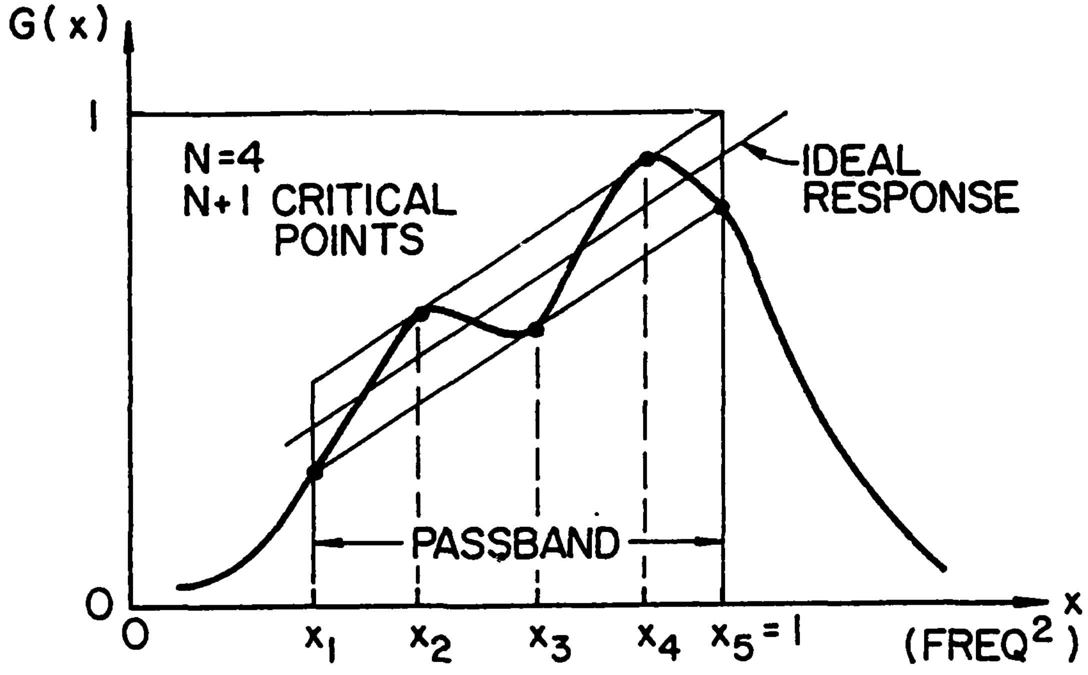


Figure 2.3. Typical Even-Order Gain Function

2.4. Odd-Order Equiripple Gain Function Synthesis

Simply extending the even-order procedure to odd N brings immediate difficulty. Were we to simply select N+1 critical points for odd N, we would be specifying an even number of points with error of alternating sign about the desired response. This would produce a gain function of odd symmetry (as shown in Figure 2.4(a) or 2.4(b), depending upon how the points are chosen), which would violate our realizability criteria for G(x), i.e. $0 \leq G(x) \leq 1 \ \forall \ x > 0$. (Only functions which satisfy this criteria produce $z_{11}(p)$ functions that allow synthesis of ladder networks with passive elements only - which is the sole case of interest.)

This problem can be avoided by specifying the approximating function with only N critical points when N is odd, thus forcing the required even symmetry in G(x) and allowing a realizable gain function to be generated. This approach produces a system of N simultaneous linear equations, which allows determination of N of the N+1 unknown coefficients of $P_N(x)$. If we choose one of the coefficients of $P_N(x)$ (arbitrarily, the constant term) to be a parameter of the synthesis (i.e., chosen initially, and not allowed to vary), then we can solve the approximation problem with an odd-order G(x).

2.4.1. Overview of Approximation Algorithm - Odd Order

The approximation problem is well formed once the bandwidth, gain slope, and minimum insertion loss of the ideal gain function as well as the allowable ripple radius of the approximation have been specified. The computational task is then to determine the necessary denominator polynomial coefficients of G(x) to obtain the specified approximation.

This problem is solved by using several iterations of a two-step procedure, once a value for the constant term of $P_N(x)$ has been chosen:

(1) Assume locations for the points of maximum error between the specified ideal response and the approximating function (the critical points), and compute required values of G(x) at these points. Use this information to specify a system of N linear

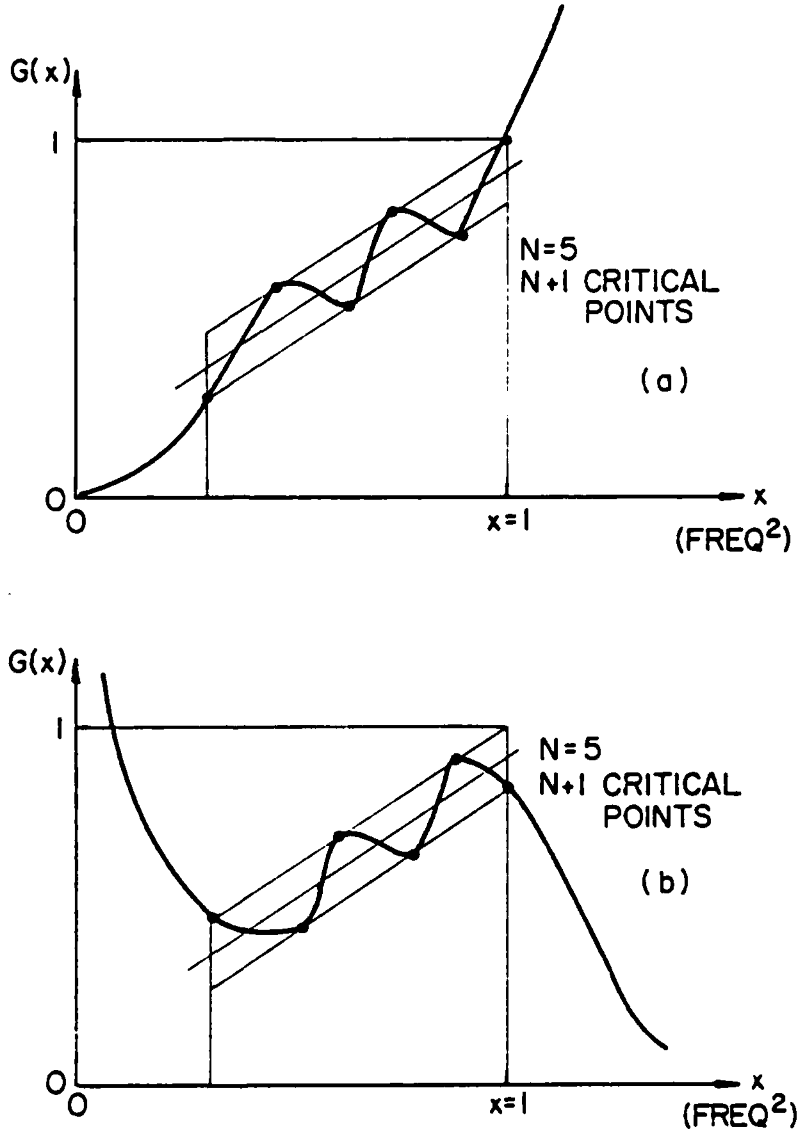


Figure 2.4. Unrealizable Odd-Order Gain Functions

equations in N unknown coefficients of $P_N(x)$; solve the system for the N coefficients.

(2) Using the $P_N(x)$ coefficients determined above, examine G(x) to determine the true locations of the critical points. Do this by solving a one-dimensional non-linear equation with multiple solutions via the Newton-Raphson Method [5]. Using these new values for the critical points, re-enter step (1).

Initially, our assumed critical points (used to solve for the coefficients) will be in error when compared to the true critical points (determined after finding the coefficients). If these new critical points are now used to again solve for the polynomial coefficients, a G(x) will be obtained whose true critical points are much closer to the values used to compute the coefficients. This iterative process converges very rapidly (usually within 4 iterations), and is terminated when the values for the critical points used in determining the coefficients are found to be within a specified tolerance of the actual values for the critical points found when subsequently examining the resulting G(x).

Besides incorporating odd-order functions, this method differs from Petersen's [1] in the method of solving the non-linear portion of the problem for the critical points. Where Petersen dealt with it as an N-dimensional non-linear problem requiring computation of a Jacobian to solve, this method treats it as a one-dimensional non-linear problem with multiple solutions.

2.4.2. Details of the Approximation Algorithm - Odd Order

The following procedure is used:

(1) Based upon specified network order and bandwidth, assume N critical frequencies (x_i, i=1,2,...,N) of alternating error (relative to the ideal function) across the passband.

(2) Compute required gain level, $G_{req}(x_i)$, at each critical point based on gain response specifications, assuming for the moment that the numerator gain constant, K, is unity. (K will be set later to insure that the maximum value of $G(x)\ \forall\ x > 0$ is 1. For the moment, no generality is lost by assuming K=1.)

(3) Now we have

$$G_{req}(x_i) = \frac{x_i^{\ k}}{P_N(x_i)}, \qquad i = 1,2,\ldots,N \qquad (2\text{-}2)$$

$$P_N(x_i) = a_0 + \sum_{j=1}^{N} a_j * x_i^{\ j}$$

$$a_0 + \sum_{j=1}^{N} a_j * x_i^{\ j} = \frac{x_i^{\ k}}{G_{req}(x_i)}, \qquad i = 1,2,\ldots,N \qquad (2\text{-}3)$$

The x_i's are assumed known at this point, so the N values for $G_{req}(x_i)$ can be computed. The coefficients a_j, j = 0,1,2,...,N are to be determined.

Equation (2-3) represents a system of N linear equations in N+1 unknowns. To allow solution of this system, a_0 is chosen to be a parameter rather than a variable, and a (positive) value for it is chosen at the start of the procedure; this reduces eq (2-3) to N equations in N unknowns.

(4) Equation (2-3) may be re-written in the form

$$X \bar{a} = \bar{b} \qquad (2\text{-}4)$$

where:

$$X = \begin{bmatrix} x_1^N & x_1^{N-1} & . & x_1^2 & x_1 \\ x_2^N & x_2^{N-1} & . & x_2^2 & x_2 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_N^N & x_N^{N-1} & . & x_N^2 & x_N \end{bmatrix} \qquad (2\text{-}5)$$

$$\bar{a} = \begin{bmatrix} a_N, a_{N-1}, \ldots, a_2, a_1 \end{bmatrix}^T \qquad (2\text{-}6)$$

$$\bar{b} = \begin{bmatrix} \frac{x_1^k}{G_{req}(x_1)} - a_0 , \frac{x_2^k}{G_{req}(x_2)} - a_0 , \quad . \; . \; . \quad , \frac{x_N^k}{G_{req}(x_N)} - a_0 \end{bmatrix}^T \qquad (2\text{-}7)$$

For best computational speed and numerical stability, eq (2-4) is solved for $\bar{a}$ by LU factorization of the X matrix [6].

factor	X	=	LU	(product of triangular matrices)
then	$LU \bar{a}$	=	$\bar{b}$	from eq (2-4)
solve	$L \bar{c}$	=	$\bar{b}$	(a triangular system)
solve	$U \bar{a}$	=	$\bar{c}$	(a triangular system)

(In the computer program, these steps are handled by calls to subroutines DGEFA and DGESL from the LINPACK subroutine library [7].)

It would also be possible to solve eq (2-4) by inverting the X matrix, and since X is of Vandermonde form, this inverse could be computed with less than the normally required amount of computation. Since it requires $3N^2$ work to invert a Vandermonde matrix and $N^3/3$ work to compute the LU decomposition, the LU decomposition is still the most efficient way to solve eq (2-4) in practical problems, however, where 6th order is the largest network usually considered.

[NOTE: At this point, the N+1 denominator polynomial coefficients of eq (2-3) have been determined. The next major task is to improve our estimate of the locations of the x_i, by examining the current G(x), and finding its true critical points. The remainder of the algorithm deals with finding these critical points.]

(5) If $G_{ref}(x)$ is the specified ideal (reference) gain function that is being approximated, then the critical points internal to the passband are those points (x_i) where

$$\frac{d}{dx}\left[G(x) \right] = \frac{d}{dx}\left[G_{ref}(x) \right] \tag{2-8}$$

The two critical points at the passband edges are fixed, and therefore known, throughout the procedure. We need to determine only the N-2 critical points internal to the passband by solving eq (2-8).

We expect to find N-2 solutions, since we have specified the approximating function to have N-2 critical points internal to the passband. By expanding eq (2-8) we obtain

$$\frac{d}{dx}\left[\ G(x)\ \right] = \frac{d}{dx}\left[\ \frac{x^k}{P_N(x)}\ \right]$$

$$= \frac{\frac{d}{dx}\left[x^k\right] * P_N(x) - \frac{d}{dx}\left[P_N(x)\right] * x^k}{P_N{}^2(x)} = \frac{d}{dx}\left[\ G_{ref}(x)\ \right] \qquad (2\text{-}9)$$

In eq (2-9) the coefficients of $P_N(x)$ are considered fixed at the values determined for them in step (4), so eq (2-9) represents a nonlinear equation in one unknown (x), with N-2 solutions. One good way to find these solutions is via the Newton-Raphson Method.

(6) We wish to find all x across the passband such that

$$\frac{d}{dx}\left[G(x)\right] - \frac{d}{dx}\left[G_{ref}(x)\right] = 0 = F(x) \qquad (2\text{-}10)$$

Equation (2-10) defines the objective function, F(x), which we will seek to minimize when finding the critical frequencies. We know that the first and second derivatives of G(x) exist because it is a rational function of polynomials; the derivatives of $G_{ref}(x)$ exist because it is an exponential function. Then, starting from an initial estimate for one of the solutions, $x_i^{(k)}$ (the i^{th} critical point), we can compute an <u>improved</u> estimate for that solution, $x_i^{(k+1)}$, via the Newton-Raphson Method and

$$x_i^{(k+1)} = x_i^{(k)} - \beta * \frac{F(x_i^{(k)})}{\frac{d}{dx}\left[F(x_i^{(k)})\right]} , \qquad \beta \le 1 \qquad (2\text{-}11)$$

Where β is a parameter used to reduce the step size in the early phases of the Newton algorithm. The update specified in eq (2-11) is iterated until some stopping criteria is satisfied.

One property of this method is its excellent convergence rate once $x_i^{(k)}$ approaches "close enough" to the true solution, x_i^*; another is its quite unpredictable performance when $x_i^{(k)}$ is "too far" from x_i^*. This second property makes it necessary to take special precautions in using this method in an algorithm to find all the solutions to eq (2-9).

(7) During the first few overall iterations of the approximation procedure, the initial estimates of the locations of the critical points are typically not very good. This means that these estimates cannot be used as reliable starting points for the Newton iterations to find the critical points for two reasons: (1) the Newton Method would sometimes converge to a point outside the desired passband, and (2) the method might sometimes "miss" a critical point by converging to a single point from two different starting points, thus producing erroneous results (all critical frequencies will not be found). An alternate strategy was devised to avoid these problems.

During the first few overall iterations, the Newton starting point for each of the N-2 critical points is chosen by a short search prior to entering the Newton iteration. During this search the second derivative of F(x) is evaluated, and a point is accepted as a starting point once the second derivative at that point is non-zero and of the proper sign for the region of the curve (G(x)) where the critical point is being sought. (This insures that the Newton iterations will proceed in the proper direction from a chosen starting point, and that we never try to evaluate eq (2-11) when the denominator might evaluate to zero.)

Depending upon the magnitude of the second derivative, and the distance away from x^*, the step size computed by eq (2-11) may be much too large. To control this problem, the update expression (based on eq (2-11)) is modified to allow a maximum step size of only 0.03 at any time. This eliminates erratic behavior of the algorithm when far from the true solution, while not hampering rapid convergence once the vicinity of the solution has been reached.

After the first few iterations of the overall procedure, the locations of the critical points change very little between iterations, and the critical points found during the previous iteration are used as the starting points for the next. This minimizes the number of Newton iterations required once the danger of erratic behavior is past.

(8) After the true critical points have been determined, the overall procedure is repeated by re-entering at step (4). Once the values for the critical points have been determined well enough that no Newton iterations are required to find any critical point after re-computing the coefficients, the loop is exited with the final values for the coefficients and the critical frequencies.

2.4.3. Implementation

The above algorithm, with very slight modification, can solve the even-order equiripple gain function synthesis problem as well. This method has been implemented in a computer subroutine for solving the lumped element approximation problem with functions of arbitrary order, and combined with other existing subroutines to form a complete lumped element network synthesis computer program (designated ARBSYN). A commented FORTRAN listing of the approximation subroutine (designated ODDSYN) is presented in Appendix A.

2.5. Effects of Varying the Denominator Polynomial Constant (DPC)

While the denominator polynomial constant (DPC) may be chosen as a synthesis parameter when designing odd-order networks, there is always a finite allowable range for DPC that will result in a realizable gain function.

The fundamental realizability criteria for G(x) is $0 \le G(x) \le 1$ $\forall\ x > 0$. We can always insure that $\max[\ G(x)\ ] \le 1$ by properly choosing the numerator gain constant, K, in eq (2-1). But the denominator polynomial coefficients control the $0 \le \min[\ G(x)\ ]$ portion of the criteria.

To have $0 \le G(x)$ for very small values of x, as $x \to 0$, we need the constant term of $P_N(x)$ to be positive. To have $0 \le G(x)$ for very large values of x, as $x \to \infty$, we need the high order coefficient of $P_N(x)$ to be positive as well.

In general, then, we can say that the (open) interval of allowable DPC values is ($0, DPC^*$), where DPC^* is the smallest positive value that produces a $P_N(x)$ with a negative high order coefficient. While the value of DPC^* varies widely with the gain function specifications being approximated, it is usually on the order of 0.01 to 1.0 for typical networks.

Any DPC value chosen from the realizable interval ($0, DPC^*$) will result in a gain function that meets the approximation specifications and allows synthesis of a physical network corresponding to that gain function, but the circuit element values and required terminating impedance can vary quite markedly as DPC is allowed to range within this realizable interval.

2.5.1. Gain-Bandwidth Behavior with DPC

One of the more important performance statistics of a matching network is its gain-bandwidth product (GBW), and for odd-order networks GBW can be observed to vary considerably with DPC. The normalized gain-bandwidth product of a network is directly proportional to the value of C_1 (the first capacitive element) when C_1 is connected in shunt, and proportional to $1/C_1$ when C_1 is connected in series. Using this, we can examine the GBW of several different networks as DPC is varied over its realizable limits.

Each of the four 3^{rd} order networks shown in Figures 2.5 and 2.6 were synthesized for 0 gain slope, 0.05 dB ripple, 0 minimum insertion loss, and octave bandwidth. In each case DPC was varied over the full realizable range, and the plots show the resulting behavior of GBW for each case. The lower limit on DPC is zero in all cases, and the maximum realizable value is determined by the procedures described in Chapter 4.

In many cases, a DPC value can be found internal to the realizable interval which maximizes GBW. Two examples of this behavior are shown in Figure 2.5. The value of DPC which maximizes GBW for one network, however, is not related to the DPC which maximizes GBW for another network, nor are the general "shapes" of the GBW vs DPC curves necessarily similar for different networks.

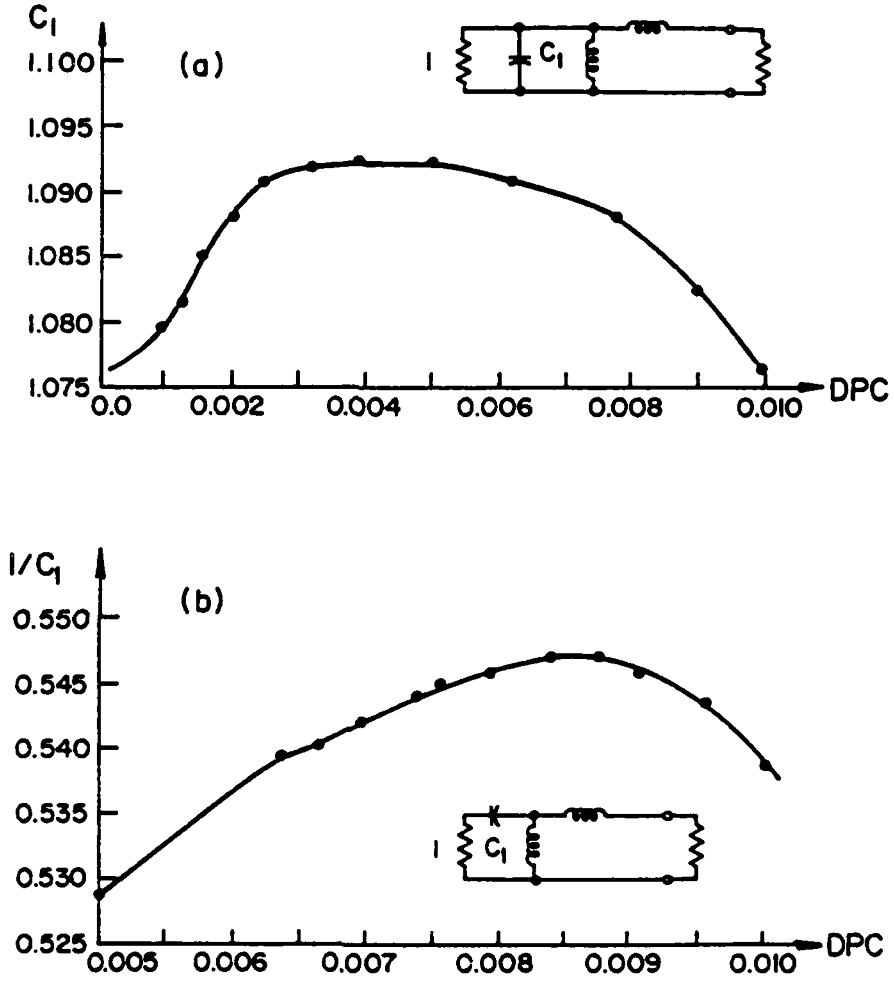


Figure 2.5. Gain-Bandwidth Effects of DPC Variation

In some cases, the DPC value that would maximize GBW falls outside the realizable DPC interval. In these cases, the DPC that "maximizes" GBW will be found to be at one of the boundaries of the realizable interval for DPC. Figure 2.6 shows two examples where this type of GBW vs DPC behavior is observed. In Figure 2.6(a), GBW can always be increased by reducing DPC, though below about 0.001 the effect is much less dramatic. In Figure 2.6(b), with a different network, GBW can be

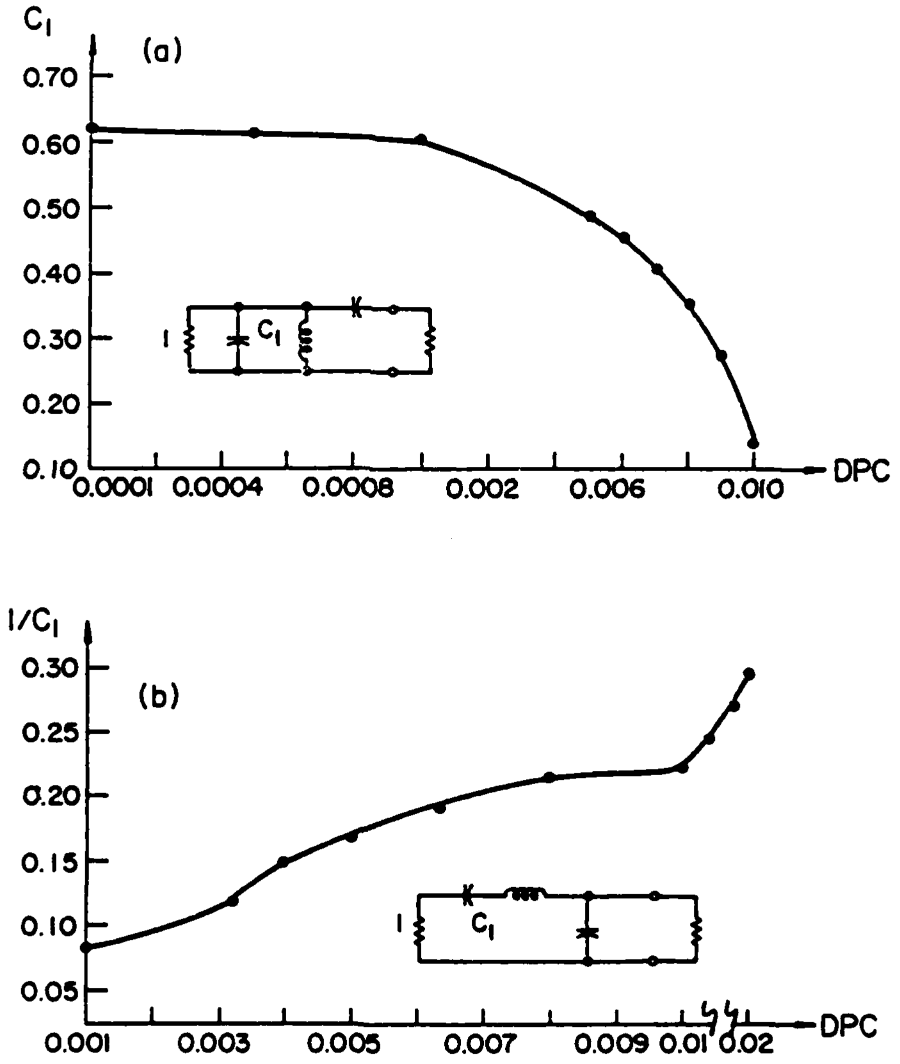


Figure 2.6. Gain-Bandwidth Effects of DPC Variation

increased by *increasing* DPC until the limiting value for realizability is encountered (at about 0.023 in this example). (Note that the DPC axis in Figure 2.6(b) is not drawn to scale.)

2.5.2. Terminating Impedance Behavior with DPC

Different values for DPC in an odd-order network produce different terminating resistance values in the synthesized network. Figure 2.7 shows two typical examples of this behavior, where both networks were designed for 0 gain slope, 0 minimum insertion loss, and 0.05 dB ripple over an octave band.

The fact that choice of DPC affects R_{out} as well as the size of the first reactive element in an odd-order matching network is important in the design of practical matching networks, where a simple 3rd order network can often be designed to absorb an FET's parasitic capacitance and provide a required impedance transformation instead of using a larger 4th order network.

2.6. Choosing DPC

The denominator polynomial constant is an additional synthesis parameter, available only when designing odd-order networks, and some discussion is warranted on how best to use it in designing circuits.

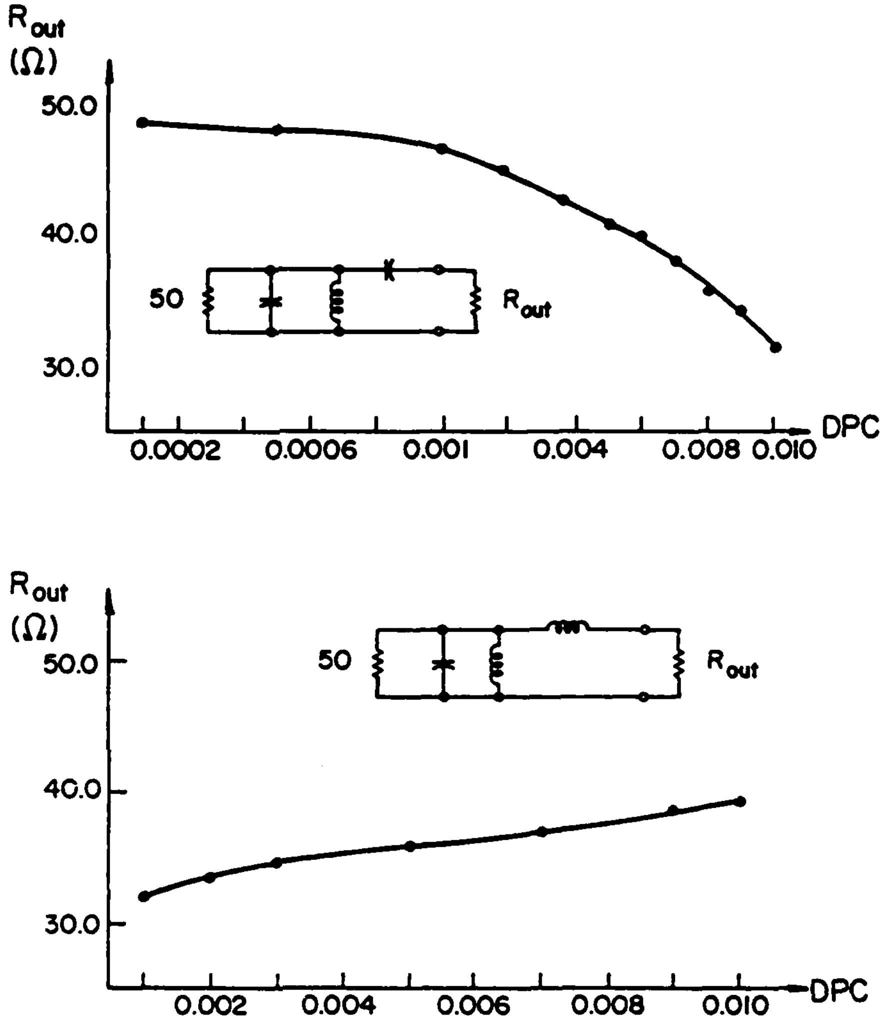


Figure 2.7. R_{out} Effects of DPC Variation

2.6.1. Simultaneous Parasitic Absorption and Impedance Matching

The most common problem in designing practical monolithic amplifiers is that of finding matching networks with a minimum number of elements that allow simultaneous absorption of an FET's parasitic capacitance and impedance matching to a specific load or source. The best strategy for this situation is to simply use DPC in the same manner that the other synthesis parameters are used to search out a combination of slope, ripple, bandwidth, minimum insertion loss, and DPC that allows solving the matching problem, preferably without requiring a Norton Transformation (see § 2.12).

In this instance, the availability of DPC as an extra synthesis parameter increases the likelihood that a suitable network of lower order can be found to solve the particular matching problem.

2.6.2. Unique DPC for Maximum Gain-Bandwidth Product

As discussed above, there is always a DPC value (either internal to the realizable interval for DPC, or at one of the endpoints) that maximizes the gain-bandwidth product of a matching network. While the optimum DPC for best GBW can be found by trial and error, a systematic search routine has been incorporated into the ARBSYN computer program to allow quick and convenient identification of the maximum gain-bandwidth solution when synthesizing odd-order networks. This routine represents a compromise between computer execution time and the accuracy with which the optimum DPC is identified. As implemented, the routine identifies the best value for DPC within 5%, and does this in 30 iterations or less.

The search algorithm is as follows:

(1) Initially examine the resulting GBW of networks synthesized with DPC values spanning the interval [0.001, 10.0]. Perform up to 10 evaluations, with DPC values spanning the interval in a geometric progression.

Save the DPC value from this sub-search (DPC_1) that produces the best GBW.

(2) Reduce the search increment, and search a much reduced interval including DPC_1 for the value that produces maximum GBW (DPC_2). Perform up to 10 evaluations.

(3) Again reduce the search increment, and search the reduced interval which includes DPC_2 until the value that produces maximum GBW (DPC_3) is found. Perform up to 10 evaluations. DPC_3 is then within 5% of the true value which optimizes gain-bandwidth product.

Within each sub-search, evaluations are performed by successively increasing DPC. Once a DPC value is reached that produces an unrealizable gain function, that sub-search is terminated and the algorithm advances to the next stage. (Once an unrealizable DPC has been encountered, all larger values will likewise be unrealizable.)

Each GBW evaluation within a search involves synthesizing a network with the current DPC value, and noting the resulting GBW value from the first (capacitive) element of the network.

Two other strategies were also tested for selecting a unique DPC, neither of which proved to be successful. The first was to choose DPC such that the <u>second</u> derivative of the gain function at the center critical point was zero; it turned out that equiripple solutions with zero second derivatives at the center critical point do not exist. The other was to choose DPC such that the slope of G(x) at the upper passband edge was made most negative, in hopes of maximizing gain-bandwidth performance; it turned out that this slope was not correlated to gain-bandwidth performance.

2.7. Practical Monolithic Input Networks

There are only a few input network topologies which need to be considered when designing practical monolithic amplifiers. Since FET bias will need to be introduced, we require that the network contain a shunt inductor as the first element at the FET's terminals, so that bias may be introduced by simply adding a DC bypass capacitor at the ground end of the inductor. We would also like the network to contain a series capacitor so that no additional DC blocking capacitor is required at the input, but this is not a necessity. We also require that the 4th order networks contain an adjacent pair of series and shunt inductors so that a Norton Transformation may be used, if needed to match impedances.

One of the three 3rd order networks shown in Figure 2.8(a-c) will often be able to handle any practical narrowband or wideband FET matching problem without requiring a Norton Transformation, but one of the two 4th order networks at Figure 2.8(d-e) will almost always be able to solve the problem when the 3rd order networks cannot. The 4th order networks will usually require a Norton Transformation to match impedances, and this can be accommodated by splitting and re-arranging the two inductors. In the figure, the assumed resistive source impedance for the amplifier is shown at the left in the networks, and the series R-C model for the input of the FET is shown at the right.

Second order networks are not practically useful in themselves, since they will almost never be able to simultaneously absorb an FET's parasitic capacitance while providing a required impedance transformation. Fifth, sixth, and higher order networks are likewise not treated

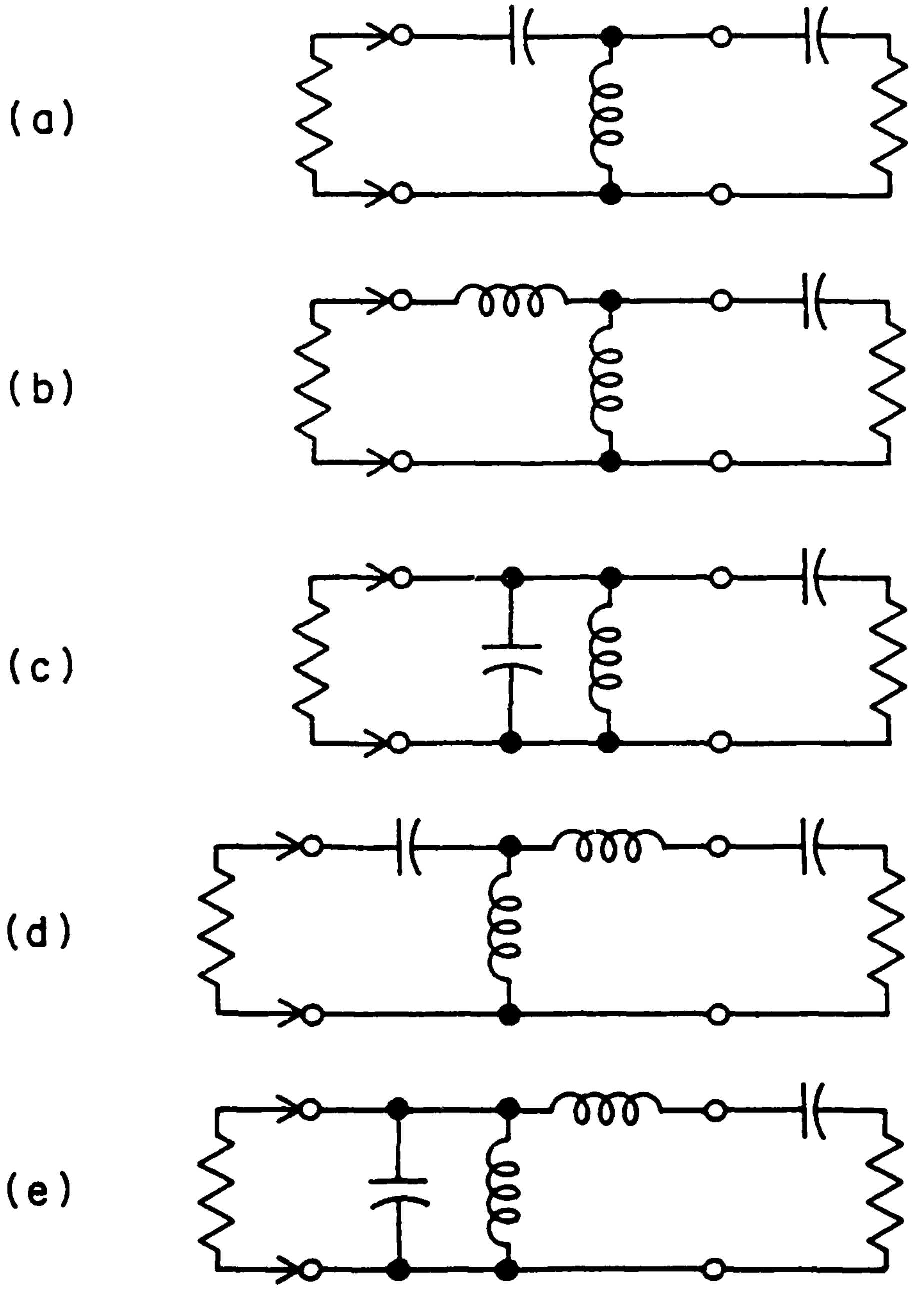


Figure 2.8. Practical Lumped Element Monolithic Input Networks

here (for input and output networks) since they contain too many circuit elements to be of practical use in MMIC's, especially GaAs MMIC's where the cost of wafer area is so high.

2.8. Practical Monolithic Output Networks

All of the practical considerations discussed above apply equally well to the choice of a monolithic output network for an FET, but the topologies are slightly different. The circuits shown in Figure 2.9(a-b) are 3rd order networks, while those at Figure 2.9(c-d) are 4th order circuits. In the figure, the shunt R-C model for the output of the FET is shown at the left, while the assumed resistive load impedance for the amplifier is shown at the right.

2.9. Practical Monolithic Interstage Networks

The task of designing a monolithic interstage network is almost always much more difficult than that of designing an input or output network. This is for several reasons: (1) the problem now requires exact parasitic absorption at both ends of the network, (2) the required impedance transformation (from the output resistance of one FET to the input resistance of another) is usually quite extreme, and (3) large values of gain slope compensation are often required.

We would again like to be able to introduce FET bias voltages with the addition of a minimal number of elements, so this dictates shunt inductors at the FET's and a series capacitor to provide DC blocking

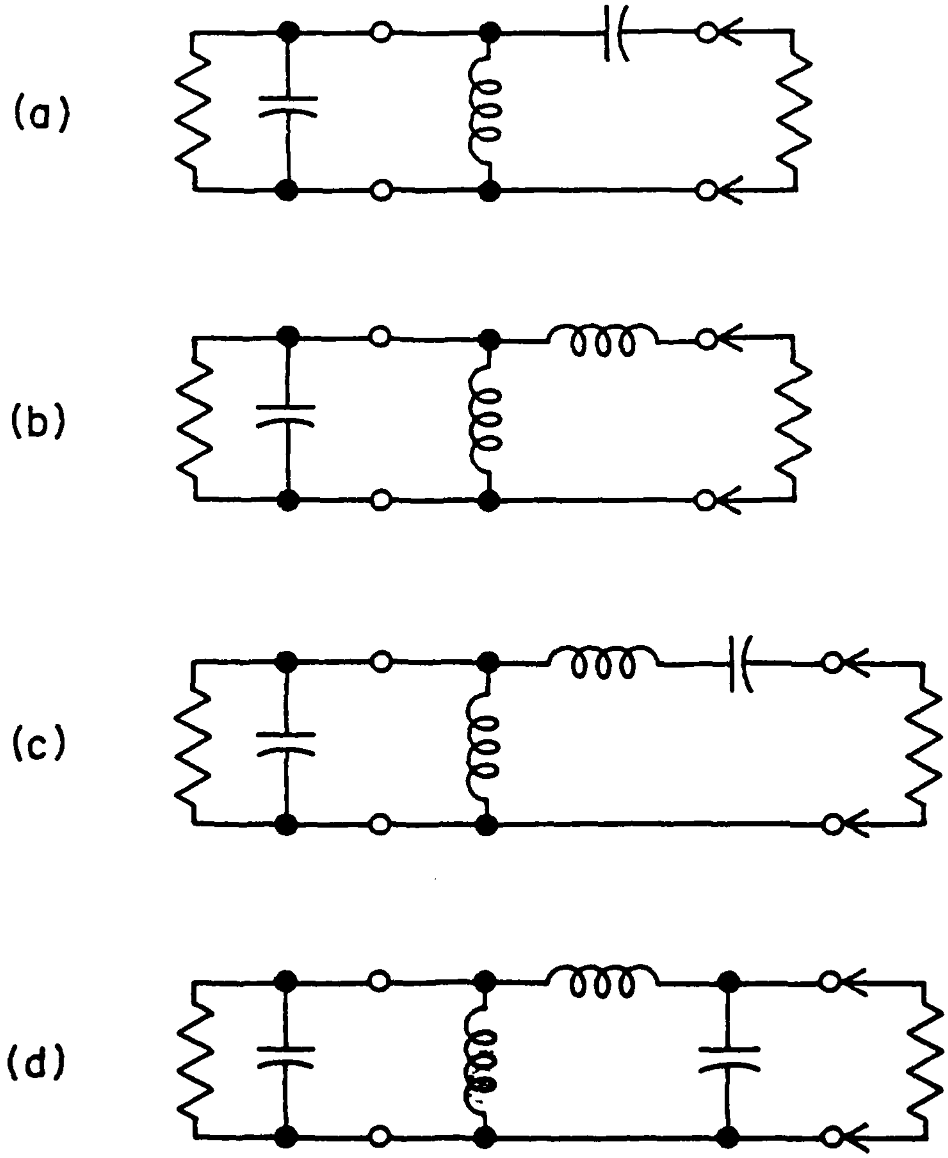


Figure 2.9. Practical Lumped Element Monolithic Output Networks

between the two stages if possible. Candidate topologies that meet these criteria are shown in Figure 2.10. (The output models for the first FET's are drawn at the left in the figure, and the input models for the second FET's are shown at the right.)

The 4th order interstage network at (a) in Figure 2.10 represents the smallest practical circuit that might be synthesized to serve as an interstage network, and it also has the advantage that a Norton Transformation is possible to assist in impedance matching. This network, however, would require a DC blocking capacitor as well as an additional RF choke (besides bypass capacitors) to by fully usable.

The 6th order interstage topology at (b) of Figure 2.10 is also a good candidate that requires only the addition of bypass capacitors to be fully usable, and it can potentially use a Norton Transformation for impedance matching.

When neither of the above networks prove to be satisfactory, there are still two more possibilities. Neither can produce a gain response whose performance specifications can be predicted exactly in advance (unlike the others), but they do present practical alternative solutions to the interstage matching problem.

In both cases the general strategy is to first form a 2nd order network at the input to the second FET by adding a shunt inductor. This network should exactly absorb the FET's parasitic input capacitance, and will produce some (uncontrolled) impedance transformation to a resistive value R^*. The interstage problem has now been reduced to one of matching the FET's output to the resistive load R^*-- an output matching problem.

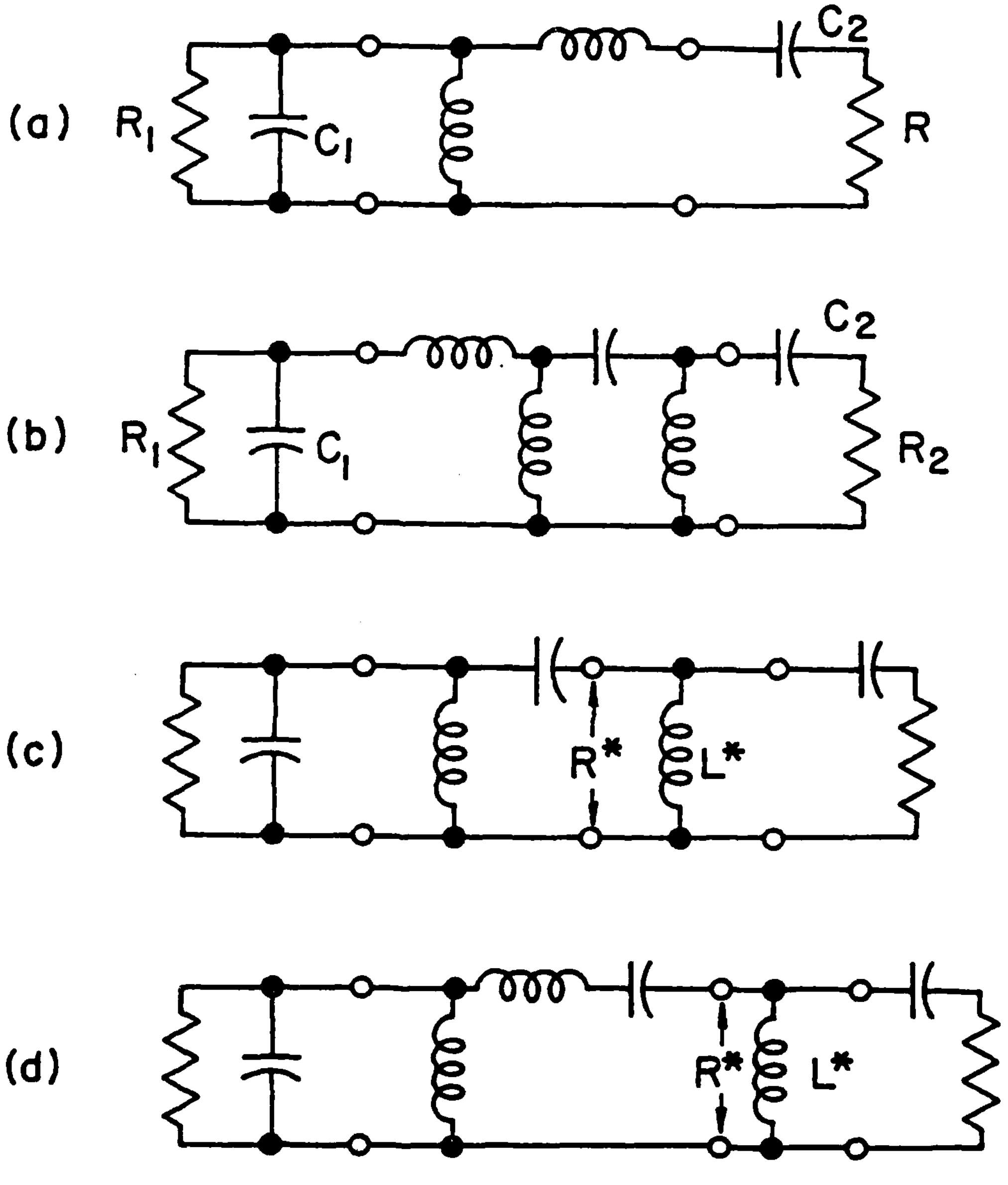


Figure 2.10. Practical Lumped Element Monolithic Interstage Networks

This part of the problem can either be solved with a 3rd order network (Figure 2.10(c)) or a 4th order network (Figure 2.10(d)), and the two networks cascaded to form the overall interstage network.

Due to the gain ripple, and perhaps gain slope, in the responses of the two networks, the overall gain response of the cascaded network over a wide band is not directly predictable from knowledge of the individual gain responses themselves. In practice, however, this is of little consequence, since computer optimization can be applied to the cascaded network to usually obtain a very satisfactory gain response even over an octave band or more.

In theory, the cascaded technique is inferior to the 4th, 5th, or 6th order directly synthesized interstage networks because: (1) it is not possible to exactly predict the overall gain response of the cascade, (2) the cascade often requires some computer optimization to obtain a satisfactory wideband gain response, and (3) the gain response of the cascade will often not be as "good" (low in ripple) as that of a corresponding directly synthesized network. In practical terms, however, the cascade technique almost always provides a solution to a difficult matching problem, even when a directly synthesized network cannot. It can therefore be very useful in designing MMIC's.

Regardless of the topology chosen, it is important that all possible distributions for the zeroes of the $s_{11}(p)$ function be examined during the search for a suitable interstage network. The default distribution (all zeroes in the left half complex plane) is often not the optimum for solving interstage matching problems.

When using topologies (a) and (b) note that during the initial phases of the synthesis the search is for a network that absorbs C_1 exactly with a source resistance of R_1, and has an R_2C_2 product the same as that for the FET input model being matched. After this condition has been achieved in a network, a Norton Transformation may be performed to adjust the output resistance value to that required by the FET model. Since the RC product is invariant under impedance scaling, the parasitic C value will also be correct once the output resistance has been adjusted.

2.10. General Odd-Order Gain-Bandwidth Results

It is interesting to investigate the general gain-bandwidth behavior of odd-order networks and relate this to the known characteristics of the gain functions.

The fundamental integral restrictions for gain-bandwidth originally derived by Bode [13] and expanded by Ku and Petersen [14] are useful for examining how gain-bandwidth behavior is related to the gain function. For the shunt RC case (low-pass (LP) constraint) this restriction is

$$\int_0^\infty \ln \left[\frac{1}{1 - G(\omega)} \right] d\omega \leq \frac{2\pi}{RC} \qquad (2\text{-}11)$$

and for the series RC case (high-pass (HP) constraint) it is

$$\int_0^\infty \ln \left[\frac{1}{1 - G(\omega)} \right] \left[\frac{1}{\omega^2} \right] d\omega \leq 2\pi RC \qquad (2\text{-}12)$$

When comparing two networks for gain-bandwidth (GBW) performance, assuming both meet the same gain envelope specifications within the passband, the "better" network is the one with the smaller value for the GBW integral (the left-hand side of eq (2-11) or (2-12), for the LP or HP cases, respectively). This generally corresponds to "steeper skirts" in the gain response outside the specified passband.

Figure 2.11(a) illustrates why 6^{th} order networks exhibit better gain-bandwidth performance than 4^{th} order networks (with both designed to the same passband specifications). The 6^{th} order response, with 3 passband ripples, has a sharper drop in gain both above and below the passband than does the 4^{th} order response with 2 passband ripples. This is the familiar case for even-order bandpass networks where increasing network order always produces a higher GBW product.

When synthesizing a 5^{th} order network to the same passband specifications, however, something slightly different occurs. Figure 2.11(b) illustrates the relative position of typical 4^{th} and 5^{th} order gain responses. Since we are still required to synthesize the 5^{th} order network with just 2 passband ripples for realizability, the typical result is a gain response superior to the 4th order case at frequencies above the passband, but worse than the 4^{th} order response below the passband.

To see the reason for this, let's look at what is happening with the denominator of the gain function (the numerators are the same in both cases). Figure 2.12 shows an approximate plot of typical 4^{th} and 5^{th} order denominator polynomial values for a gain function of the type shown in Figure 2.11(b). At this time it is also helpful to recall the form of

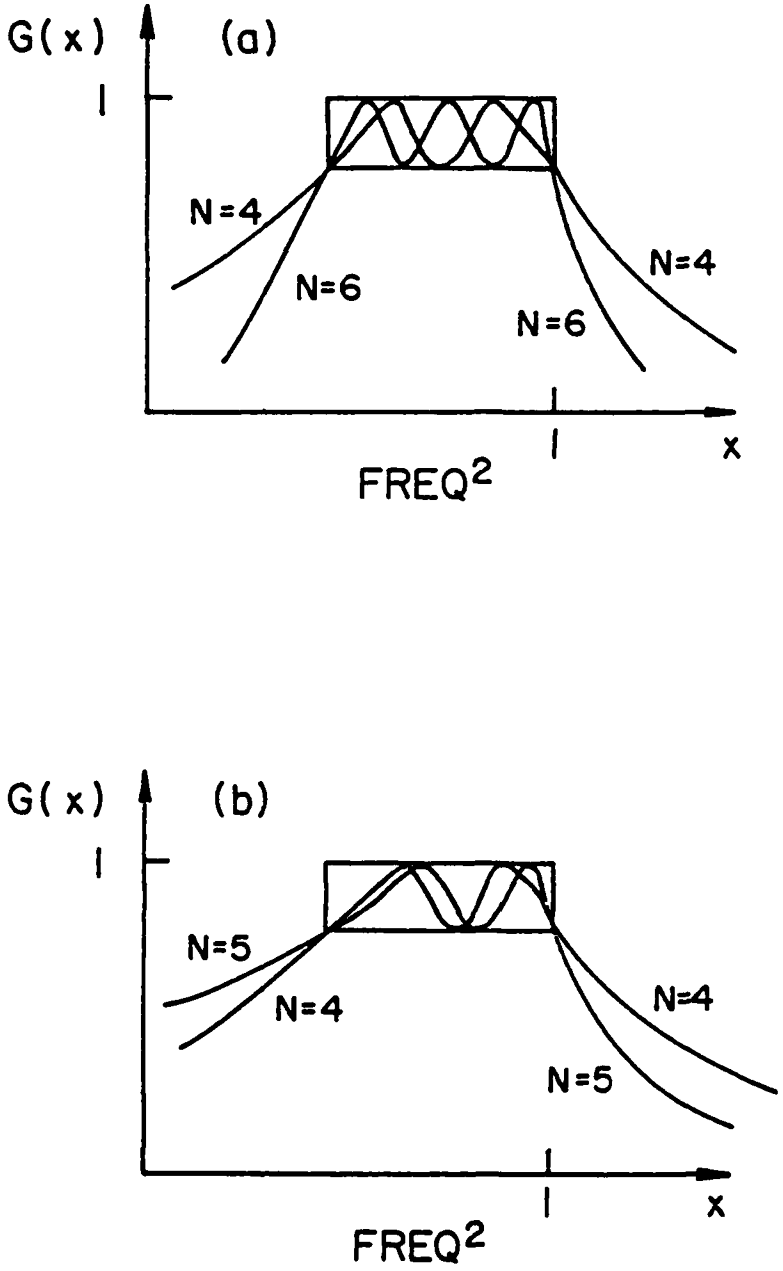


Figure 2.11. Gain Responses Outside the Passband

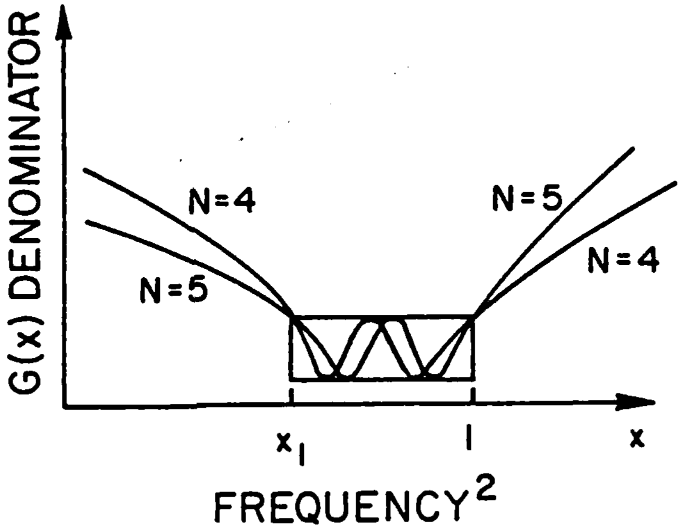


Figure 2.12. Typical Gain Function Denominator Values

the 5th order gain function ($x=\omega^2$ and k=number of high pass elements)

$$G(x) = \frac{x^k}{\alpha_5 x^5 + \alpha_4 x^4 + \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0}$$

Above the passband ($x > 1$) the 5th order denominator term causes the denominator to grow positively at a greater rate than that achieved by the 4th order function -- this is expected. Below the passband ($x < x_1$), however, the 5th order denominator cannot grow as rapidly as the 4th order function. This is because since the 5th order function has odd symmetry, it naturally tends toward negative values below the passband since we synthesize it to tend toward increasingly positive values above the passband.

We can control this tendency somewhat by forcing the ordinate intercept of the 5th order function to occur at some positive value (by choosing DPC, or α_0, to be positive), but this intercept will always be at or below the intercept of the corresponding 4th order function. The result of this when evaluating the gain response is that, below the passband, the 5th order gain response will always be above that of the corresponding 4th order response.

There are two limiting cases where the 5th order performance approaches that of the 4th order network. If α_0 is chosen so that α_5 is extremely small, the denominator becomes effectively a 4th order function, and the "5th order" gain response approaches that of the 4th order network both above and below the passband. If α_0 is chosen to be very small (but positive) then the 5th order response approaches that of the 4th order network with k-1 high-pass elements.

With this understanding of the odd-order gain function characteristics we can again return to the issue of gain-bandwidth performance. For the low-pass case, where the integral relation of eq (2-11) applies, we anticipate that the superior performance of the 5th order network in the very large region above the passband ($1 < x < \infty$) will outweigh the effect of the inferior performance below the passband to result most often in the 5th order network's GBW performance surpassing that of the corresponding 4th order network in the low-pass constraint case. We observe, however, that it is conceivable that for some combinations of gain function specifications and α_0 it might be possible to obtain 5th order GBW performance inferior to that for the 4th order network.

In the high-pass constraint case, the integral relation of eq (2-12) applies, and we notice that a ($1/\omega^2$) weighting function has now been included. This greatly increases the significance of the gain curve for very small values of ω (the region below the passband), and we observe that this greatly increases the possibility of the GBW performance of a 5th order network being worse than that of the 4th order network -- for the HP constraint case. In practice, we find that this indeed occurs, with the best GBW performance of a 5th order network with a HP constraint often not exceeding that of the corresponding 4th order network.

This discussion applies equally well to the 3rd order case as well as to all other odd-order networks synthesized via this technique of using N critical points in the approximation. In the 3rd order case, however, the improvement in the response above the passband usually outweighs the poorer response below the passband -- even in the HP constraint case -- so that a 3rd order network optimized for gain-bandwidth will usually be able to exceed the GBW performance of a corresponding 2nd order network regardless of gain function specifications.

In summary, since the gain response of an Nth order odd network does not fall off as rapidly as that of an N-1 order even network in the region below the passband, and since the gain-bandwidth product is a weighted integral over all positive x, the Nth order odd network will in some cases be able to achieve gain-bandwidth performance superior to that of a corresponding N-1 order even network, but in some cases it will not. The odd network will always, however, be capable of at least equalling the GBW performance of the even network. In addition, it is sometimes possible to obtain an Nth order odd network with GBW performance inferior to that of an N-1 order even network -- this is especially true in the high-pass constraint case with low gain slope.

2.11. Specific Odd-Order Gain-Bandwidth Results

This section presents the results of specific studies into the gain-bandwidth performance of 3rd and 5th order networks compared to that of even-order networks with identical gain function specifications. Cases were examined with 0 and 6 dB per octave gain slopes, narrow and wide bandwidths, and low-pass as well as high-pass reactive constraints. The plots illustrate how gain-bandwidth varies with the ripple specification for networks of different order.

The odd-order curves in all plots show only the performance when DPC is chosen to maximize gain-bandwidth. In all odd-order cases it is also possible to achieve poorer GBW performance by choosing non-optimum DPC values, if that is desired. The GBW performance of the Nth order odd networks can always be reduced to that of the corresponding N-1 order even network by appropriate choice of DPC, and in the zero gain slope cases it is also usually possible to obtain GBW performance inferior to the N-1 case as well.

Figure 2.13 shows the comparative gain-bandwidth (GBW) performance of the even- and odd-order networks for the low-pass constraint case (first reactive element is a shunt capacitor). All networks examined for

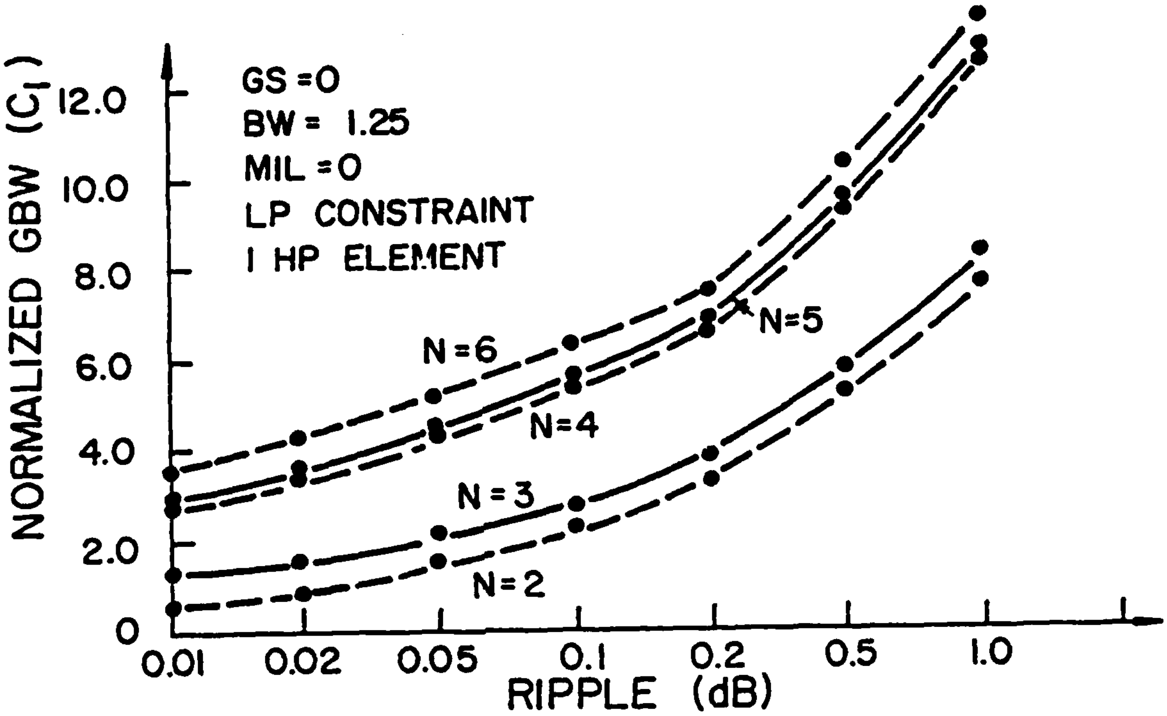


Figure 2.13. Gain-Bandwidth Results (GS=0, BW=1.25, LP)

this case were synthesized for 0 minimum insertion loss (MIL), 0 gain slope (GS), a bandwidth ratio (BW) of 1.25, and 1 high-pass (HP) element. The plotted GBW data represents the normalized size of the first (shunt C) element in the networks.

For this case the odd-order networks performed as would be expected from past results with even-order networks. The GBW curve for the 3rd order lied between the 2nd and 4th order curves, and the 5th order curve fell between the 4th and 6th order curves.

Figure 2.14 shows the comparative GBW performance for the high-pass constraint case (first reactive element is a series capacitor). All these networks were synthesized for GS=0, MIL=0, BW=1.25, and 2 HP elements. (The 6th order curve could not be extended to higher ripples since it was impossible to generate a valid gain function that met those gain specifications with the specified topology.)

The 3rd order network performance was better than that of the 2nd order network, but the 4th and 5th order networks exhibited identical GBW

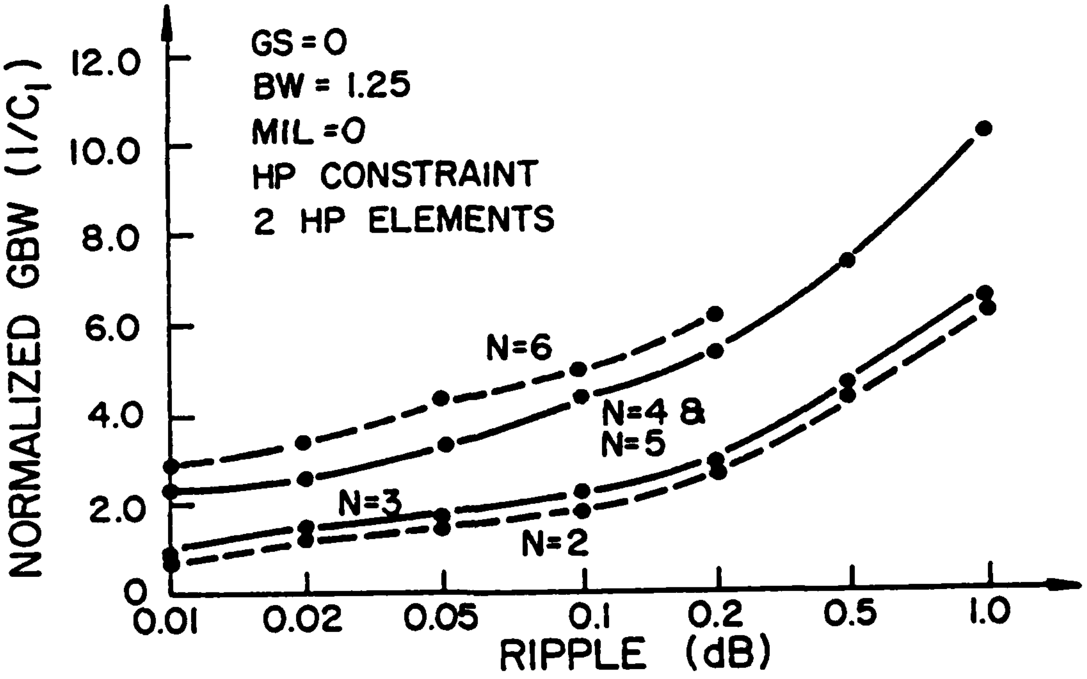


Figure 2.14. Gain-Bandwidth Results (GS=0, BW=1.25, HP)

performance. The $s_{11}(p)$ functions derived from the 4th, 5th, and 6th order gain functions in this case were examined in detail to illustrate why the 5th order GBW was approximately the same as that for the 4th order network. Table 2-I lists the complex roots of the numerator and denominator polynomials of the $s_{11}(p)$ functions for each of the three networks in question for the 0.05 dB ripple case.

Comparing first the N=4 and N=6 root patterns (cases where there was a significant difference in GBW performance), we can see that there are 2 complex conjugate pairs of roots for both numerator and denominator in the N=4 case, while there are 3 root pairs for each in the N=6 case. Furthermore, none of the 6th order roots lie very close to any of the 4th

Table 2-I. Roots of $s_{11}(p)$ Function

RIP = 0.05 dB GS = 0 BW = 1.25 MIL = 0 HP Constraint

	Numerator		Denominator	
N=4	-0.002132750	±0.8264694	-0.1003646	±0.7593087
	-0.002497883	±0.9679665	-0.1368710	±1.0354980
N=5	-0.002139014	±0.8265580	-0.1008672	±0.7592448
	-0.002490566	±0.9680869	-0.1360134	±1.0358420
	-4.557193000	0.0	-4.5572740	0.0
N=6	-0.001053324	±0.8128668	-0.0459729	±0.7797239
	-0.002188380	±0.8914190	-0.0973306	±0.8964114
	-0.001123381	±0.9862027	-0.0503410	±1.0210770

order roots in the complex plane. We also know that the in-band gain response of the 6th order network is quite different from that of the 4th order response, since the 6th order case was approximated with 7 critical points, and the 4th order case with 5.

In contrast, both the 4th and 5th order gain responses were approximated with 5 critical points, and so are almost identical within the specified passband. In Table 2-I we can see that in addition to the two complex conjugate pairs of roots in the numerator and denominator of the 4th order function, the 5th order function only adds 1 real root to each. And, significantly, the complex conjugate roots in the 5th order case have been displaced only very slightly from their positions in the 4th order case. Table 2-II quantifies this almost imperceptible change in complex root positions between the 4th and 5th order cases.

Table 2-II. Percent Change in $s_{11}(p)$ Root Positions, N=4 vs N=5
(Table 2-I Functions)

	Real Part	Imaginary Part
Numerator	0.294 % 0.294	0.0108 % 0.0124
Denominator	0.501 % 0.631	0.0084 0.0332

This represents an example of the general case discussed in the previous section where under the high-pass reactive constraint the best 5th order GBW performance is sometimes achieved by approximating a 4th order response.

Figures 2.15 and 2.16 show the GBW results for 25% bandwidth networks with 6 dB per octave gain slope. We expect the gain-bandwidth products of these networks to be consistently greater than those of the previous discussion due to the addition of gain sloping, and this is indeed the case. Once again, the general gain-bandwidth trends for the odd-order networks are as expected.

The gain-bandwidth results for octave band designs with zero gain slope are shown in Figure 2.17 (low-pass constraint) and 2.18 (high-pass constraint). Once again, the 4th and 5th order results in the HP case were identical, as they were for the 25% bandwidth case.

Figures 2.19 and 2.20 show GBW results with networks designed for 6 dB/octave gain slope and octave bandwidth. Several of the networks exhibited gain overshoot outside the specified passband for the smaller ripple values; the GBW results for these cases cannot be properly compared to the MIL=0 results for the other networks, so GBW results for these overshoot cases are not included in Figures 2.19 and 2.20. Figure 2.19 shows the results for the low-pass constraint case.

Figure 2.20 shows the GS=6, BW=2.0 results for the high-pass constraint case. For this case the odd-order networks performed as expected in comparison to the even-order networks.

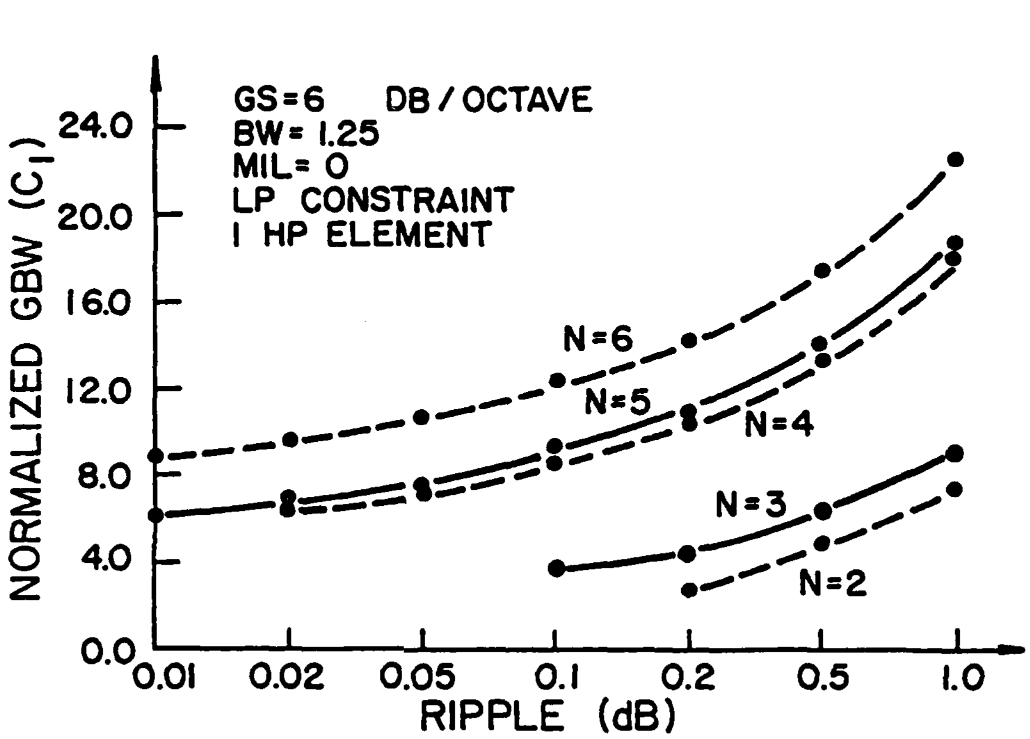


Figure 2.15. Gain-Bandwidth Results (GS=6, BW=1.25, LP)

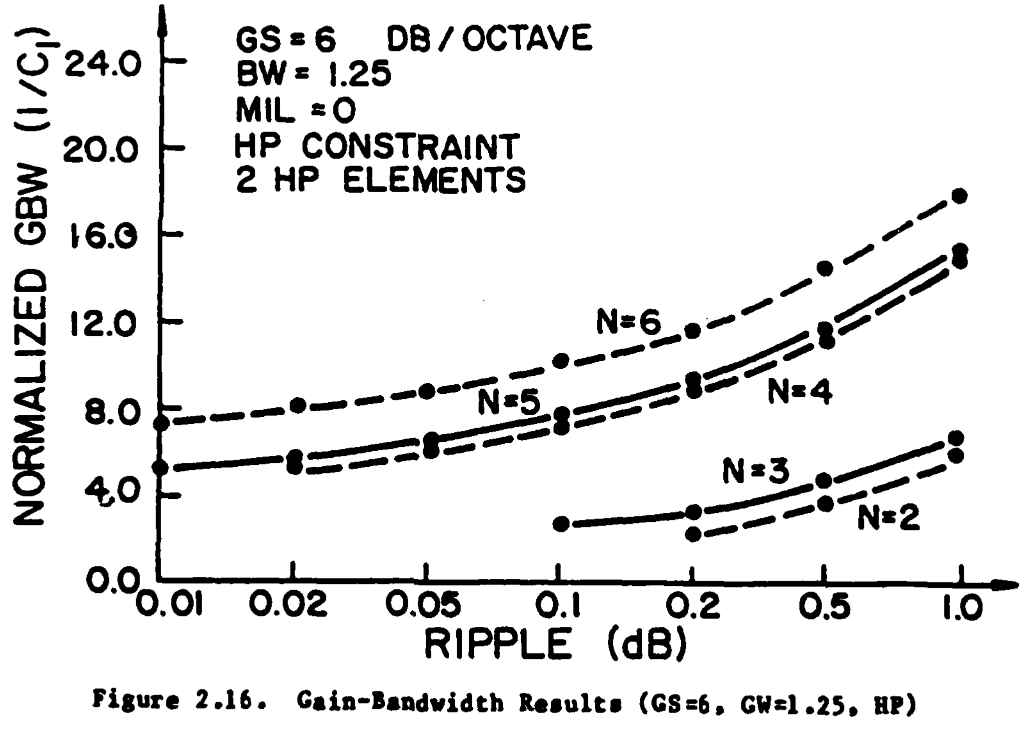


Figure 2.16. Gain-Bandwidth Results (GS=6, GW=1.25, HP)

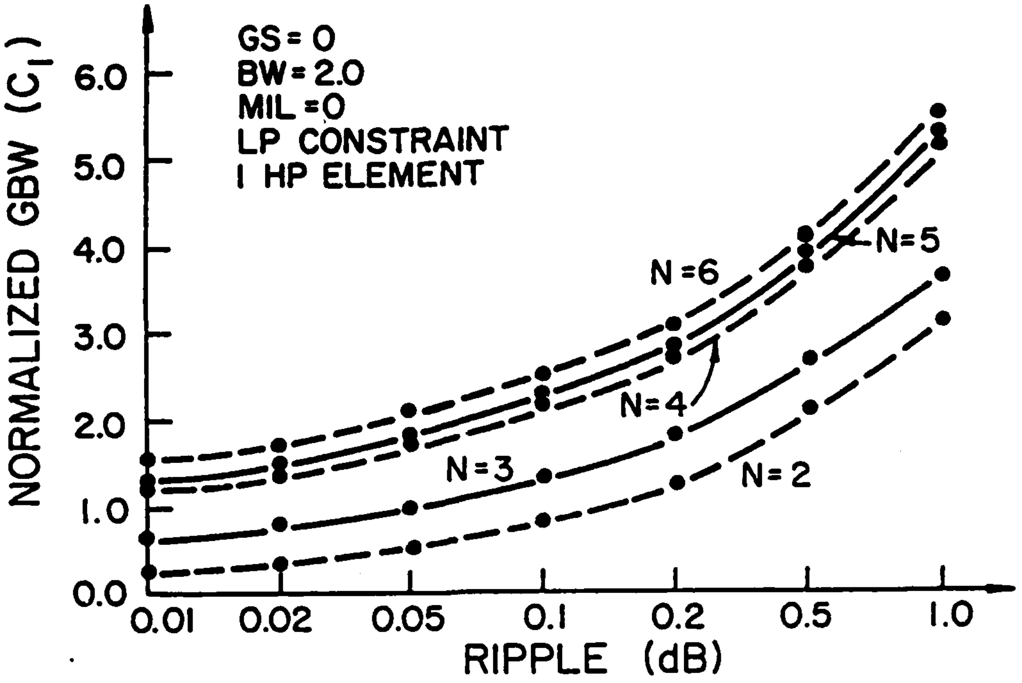


Figure 2.17. Gain-Bandwidth Results (GS=0, BW=2, LP)

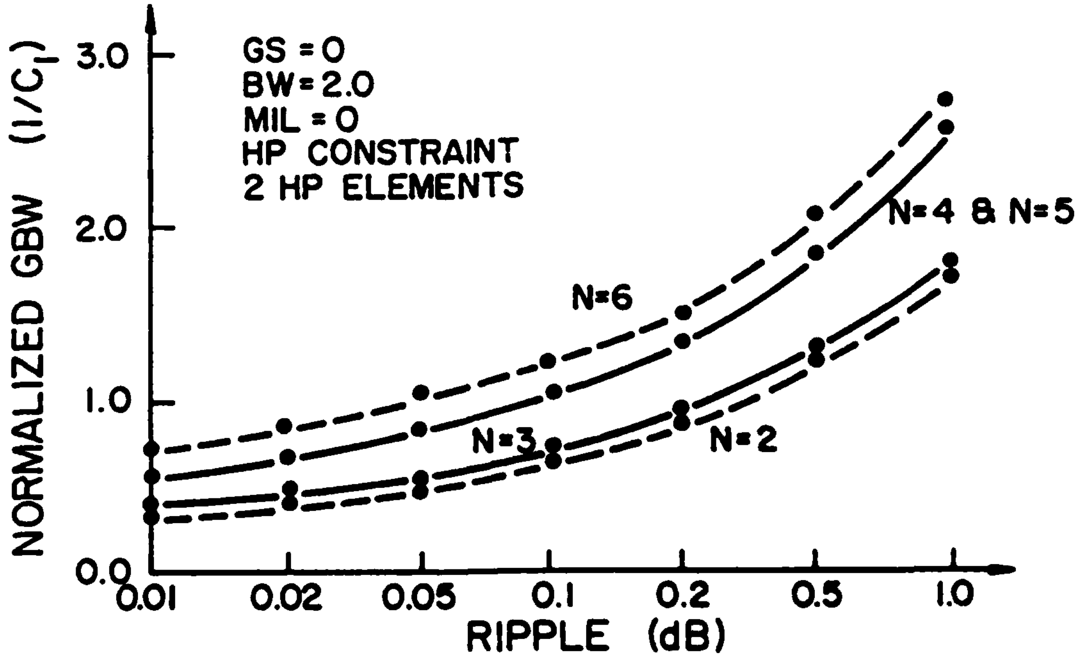


Figure 2.18. Gain-Bandwidth Results (GS=0, BW=2, HP)

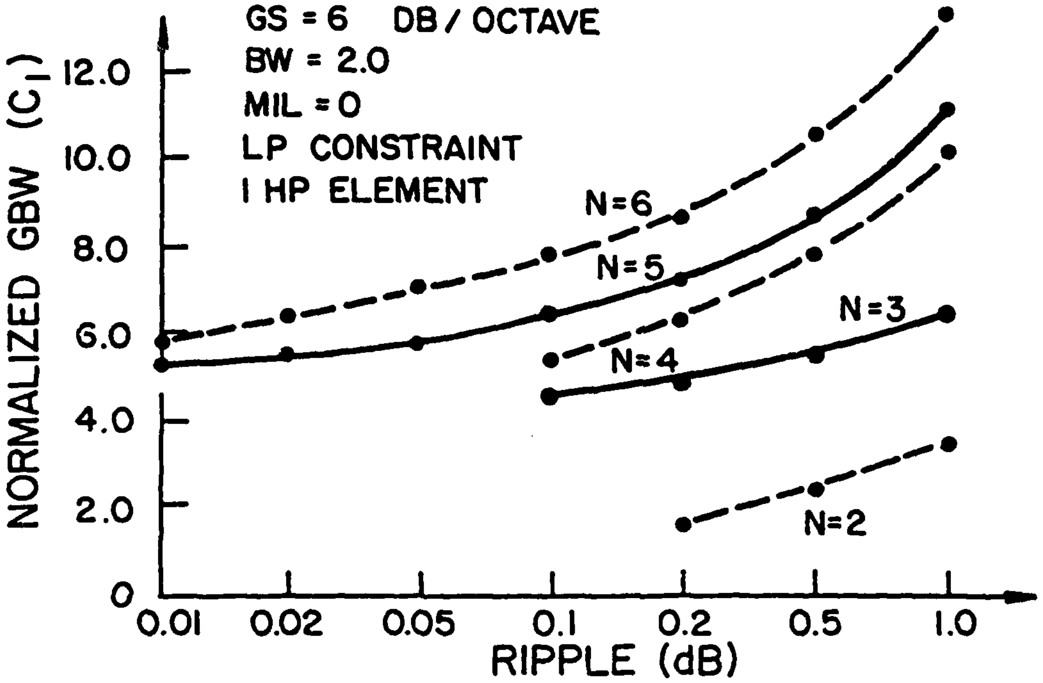


Figure 2.19. Gain-Bandwidth Results (GS=6, BW=2, LP)

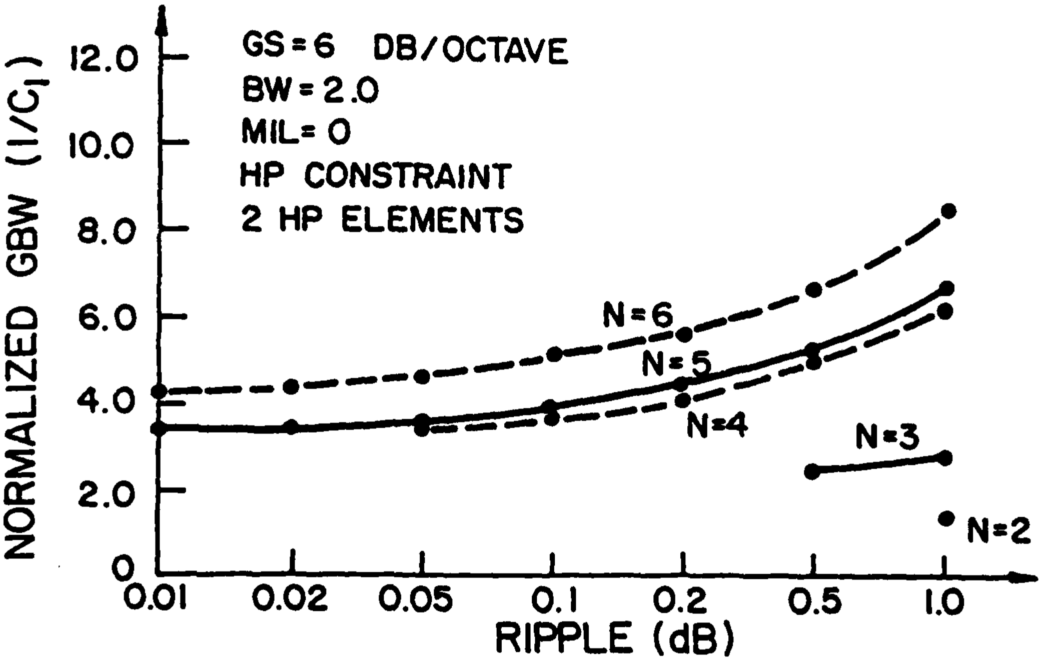


Figure 2.20. Gain Bandwidth Results (GS=6, BW=2, HP)

Another aspect of gain-bandwidth performance is that the GBW product of a network should increase as the gain reduction (MIL) in the response is increased. To illustrate this, the ripple, bandwidth, and gain slope specifications are held fixed while synthesizing networks with a varying MIL specification and noting the resulting GBW products.

Figure 2.21 shows the GBW results as MIL is varied for 2nd through 6th order networks with 0.10 dB ripple, zero gain slope, and octave bandwidth. The low-pass constraint case is shown in Figure 2.21, and the odd-order data plotted is that for the optimum GBW DPC values. The data shows that the performance of the odd-order networks follows the same pattern as that of the even-order networks, which is as we would expect.

Figure 2.22 shows the GBW results as MIL is varied for the high-pass constraint case, with the odd-order data again showing the optimum GBW DPC results. The odd-order networks again follow the same general pattern as the even networks, and we notice that, just as in Figure 2.14 where ripple is being varied while MIL is held fixed, the GBW products of the 4th and 5th order networks are the same.

In summary, the gain-bandwidth performance of odd-order networks follows the same pattern as that of even-order networks with respect to variations in the ripple and minimum insertion loss specifications. While an N^{th} order odd network usually exhibits gain-bandwidth performance somewhat better than a corresponding N-1 order even network, in some cases (especially with the high-pass reactive constraint and low gain slope), the odd network will, at best, only equal the gain-bandwidth performance of the even network.

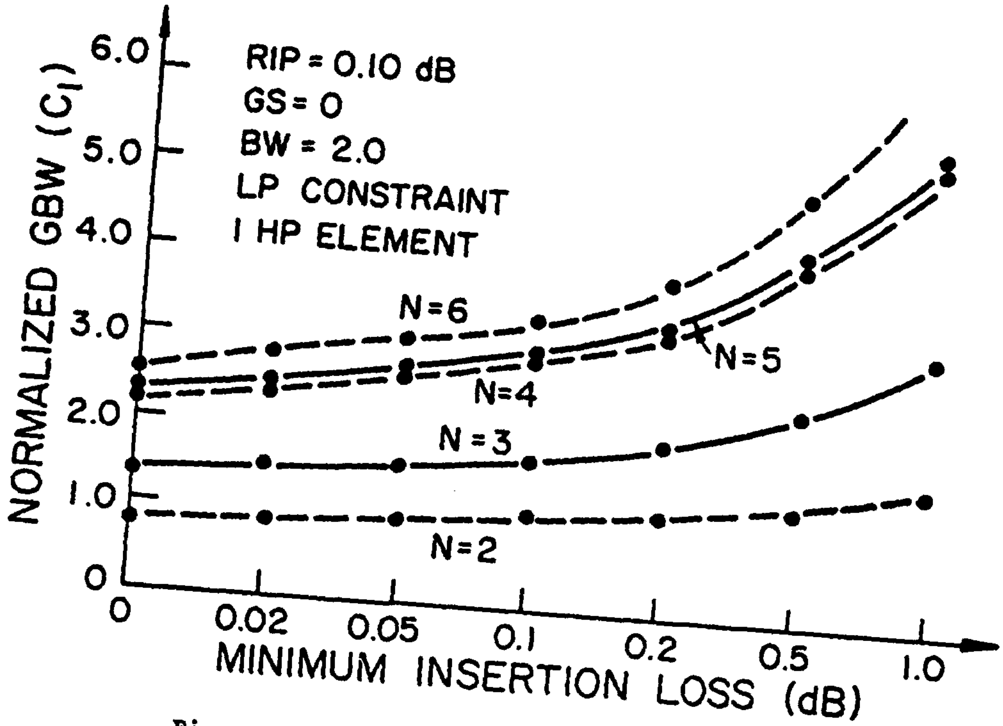


Figure 2.21. Gain-Bandwidth vs MIL (LP)

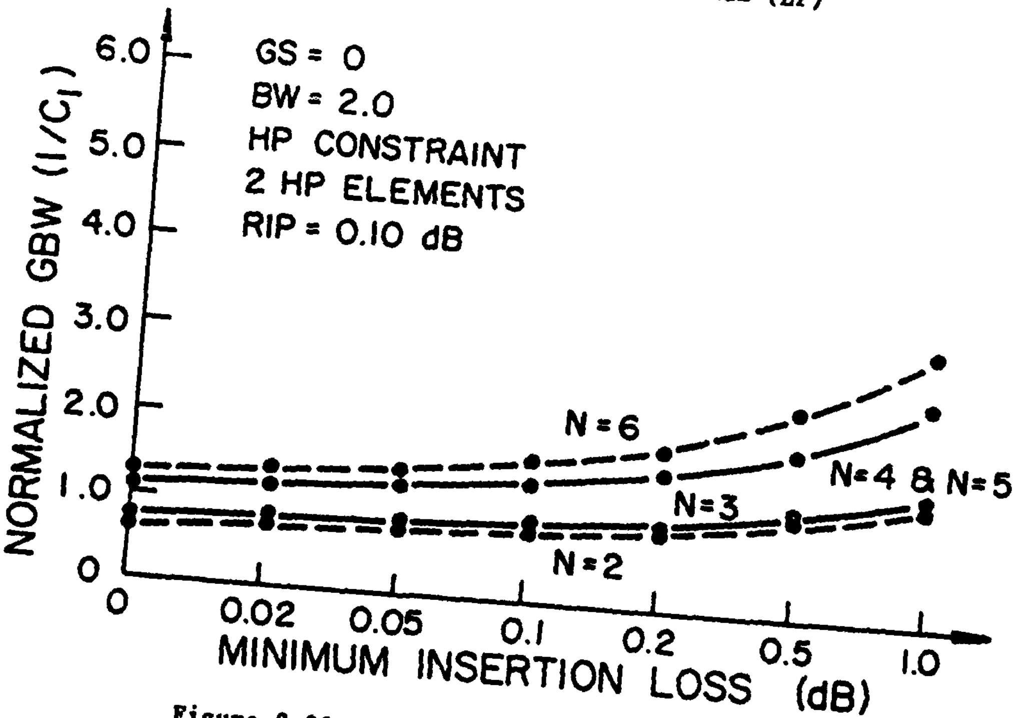


Figure 2.22. Gain-Bandwidth vs MIL (HP)

2.12. Norton Transformation

For those instances where a 3rd order network cannot be found to exactly absorb an FET's parasitic capacitance while providing a proper impedance transformation, it may be necessary to choose a 4th order topology that allows a Norton Transformation [8] to solve the problem.

When a Norton Transformation is to be used, the strategy is to synthesize the matching network so that it absorbs the FET parasitic capacitance exactly, but without regard to the required terminating impedance. Then, a Norton Transformation (Figure 2.23) is performed on two of the circuit elements to adjust the terminating impedance to 50 ohms (or some other desired level).

While there are occasions when, due to combinations of circuit element values and size of the impedance adjustment required, the Norton Transformation cannot provide the needed impedance transformation, in many cases it provides a very convenient method to adjust the output impedance of a matching network.

Figure 2.23(a) represents an initial network with an unsatisfactory output impedance -- here it has been drawn with an ideal transformer to bring the impedance up to the desired R ohms. Since the network contains adjacent series and shunt impedances of the same type (usually inductors, but capacitors would work as well), we can perform a Norton Transformation to eliminate the ideal transformer. The procedure is to "split" Z_1 into two parts (Z_{1a} and Z_{1b}) such that

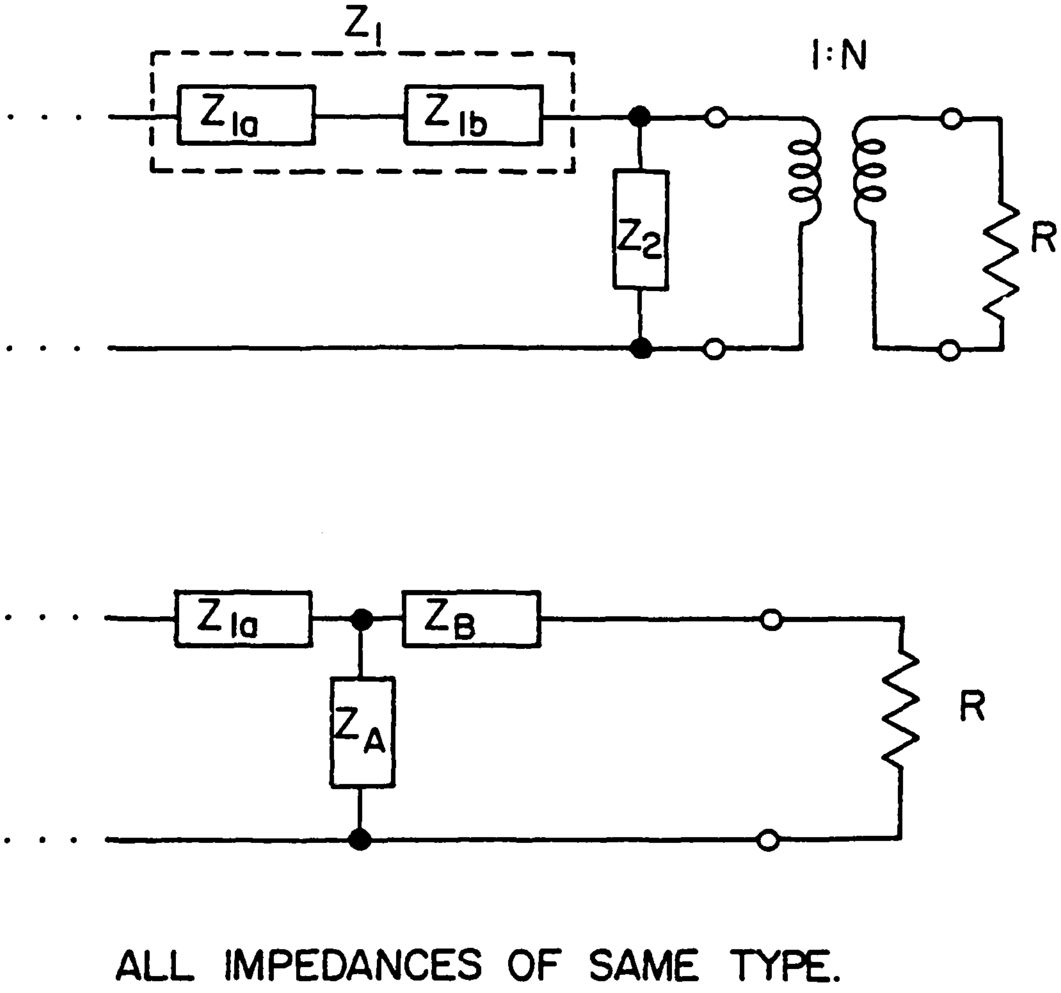


Figure 2.23. Norton Transformation

$$N = \frac{Z_{1b} + Z_2}{Z_2}$$

where N is the turns ratio of the ideal transformer to be eliminated from the network. Then by changing the topology to that shown in Figure 2.23(b) where

$$Z_A = Z_{1b} + Z_2$$

and

$$Z_B = \frac{Z_{1b} * (Z_{1b} + Z_2)}{Z_2}$$

the transformer may be eliminated from the circuit.

2.13. 3rd Order vs 4th Order Network Performance

A 2nd order network, consisting of a single inductor connected at the input or output of an FET, will almost never be capable of simultaneously absorbing an FET's parasitic capacitance while providing the proper impedance transformation to match the device to its load. In the past, this has meant that the simplest practical matching network that could be synthesized was of 4th order.

Now that odd-order networks can be synthesized, it is usually possible to simultaneously absorb FET parasitics and provide required impedance transformations with a 3rd order network, using just 1 inductor and 1 capacitor, rather than using a 4th order network requiring 3 inductors and a capacitor. The reduction in required chip area is very dramatic for monolithic realizations, and for most applications the performance of the 3rd order networks will be quite acceptable.

In the examples that follow, the 3rd order networks were designed by varying both the ripple specifications and the denominator polynomial constant (DPC) until a combination was found that simultaneously absorbed the FET parasitic and produced the needed impedance transformation to match the FET to a 50 ohm source and load. The 4th order networks were designed by varying the ripple and bandwidth specifications until a combination was found that absorbed the FET parasitics, and then applying a Norton Transformation to the inductors to adjust the terminal impedance to 50 ohms. (In one case the Norton Transformation was not necessary.)

2.13.1. Low-Noise Amplifier Example

This section will compare the performance of 3rd and 4th order matching networks used in low-noise amplifiers (LNA's) by presenting actual input and output matching network designs for a 10.4 - 12.4 GHz LNA using the Cornell 513-4 0.5 μm low-noise FET.

Figure 2.24(a) shows the input model of the device, optimized for best noise match over 10.4 - 12.4 GHz. The matching networks shown in Figure 1 both exactly match this model to 50 ohms. Both networks were synthesized for 1 dB per octave gain slope (GS) and a minimum insertion loss (MIL) of 0 dB. While the 4th order (N=4) design provides a good match over a wider bandwidth (BW) than does the 3rd order network, this extra bandwidth is not needed in this application. The 4th order design also, however, requires 2 more inductors than does the 3rd order network,

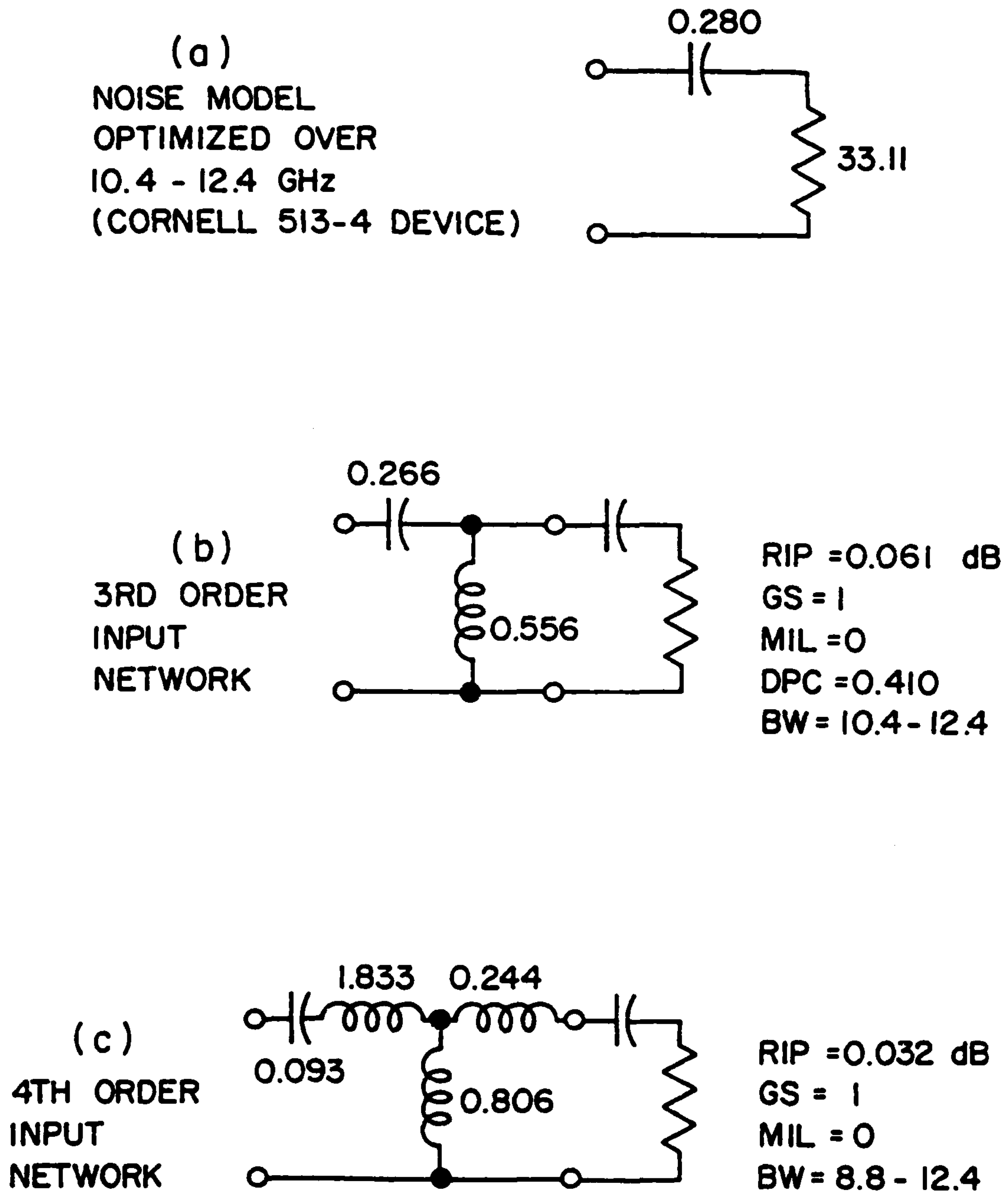


Figure 2.24. Low-Noise Amplifier Input Networks

and one of these is quite large for 12 GHz operation (1.8 nH).

Figure 2.25(a) compares the insertion loss of the two LNA input networks (analyzed with the specified model), and Figure 2.25(b) compares the input noise figure obtained with the two networks. These results show that the performance of the 3rd order network is quite comparable to that of the 4th order network over this 17% band, and that at its worst point the 3rd order design only differs from the 4th order design by about 0.1 dB in both insertion loss and noise figure.

Figure 2.26(a) shows the output model of the 513-4 device (with the network of Figure 2.24(c) connected at its input). Again, both output networks in Figure 3 were synthesized to match this model to 50 ohms. Once again, the 4th order network provides a match over a wider bandwidth than needed, and requires two more (fairly large) inductors than does the 3rd order design.

Figure 2.27 compares the insertion loss performance of the two output networks, and the 3rd order network again performs comparably with the 4th order network, but with somewhat greater ripple apparent in the response.

2.13.2. Power Amplifier Example

In general, the matching problems presented by a power FET are quite different from those of a low-noise FET, due to the very different resistance and reactance values required to model these physically very different devices. For completeness, this section will compare the perfor-

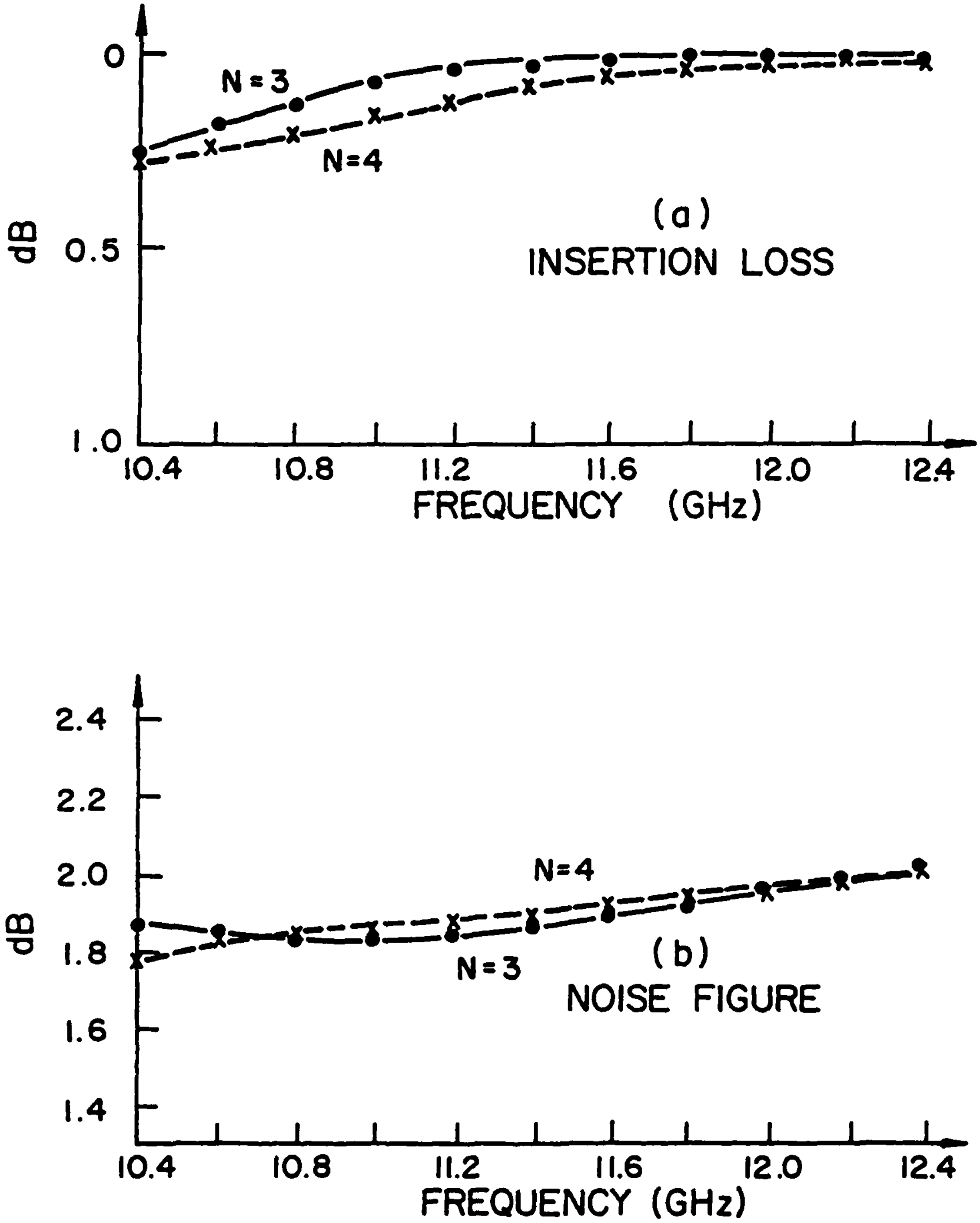


Figure 2.25. Low-Noise Amplifier Input Network Performance

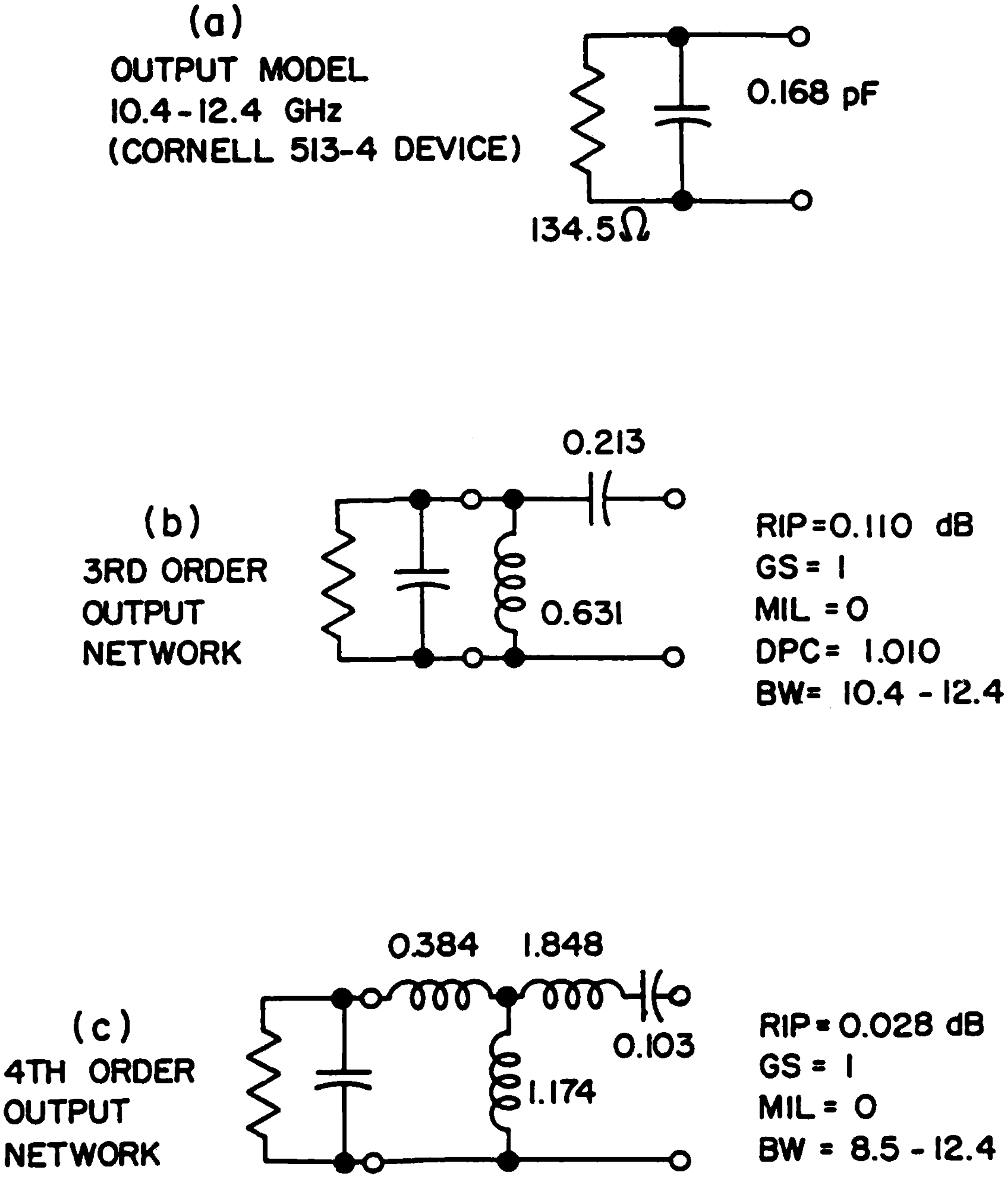


Figure 2.26. Low-Noise Amplifier Output Networks

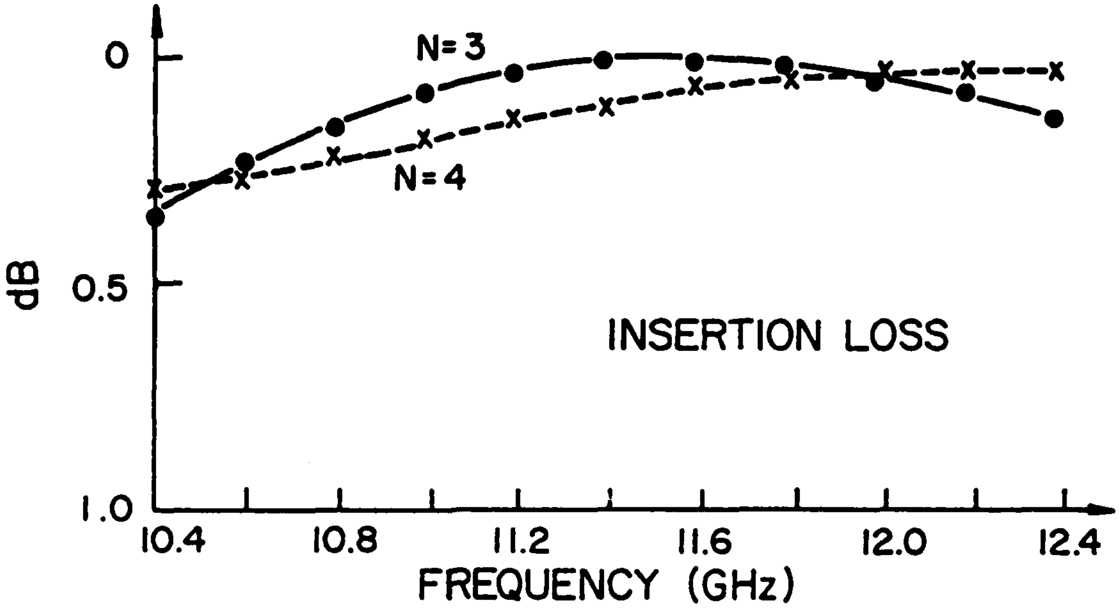


Figure 2.27. Low-Noise Amplifier Output Network Performance

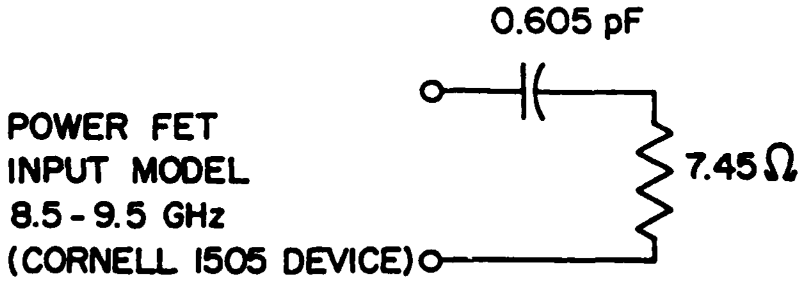


Figure 2.28. Power FET Input Model

mance of 3rd and 4th order input and output networks when designed for use with a high-power FET amplifier (using an output model for the device obtained under large signal conditions).

Figure 2.28 shows the input model of the Cornell 1505 0.8 μm power FET, optimized over 8.5 - 9.5 GHz for best match to the measured RF data. Both input networks of Figure 2.29 match this model to a 50 ohm source with 6 dB per octave gain slope. All elements of the 3rd order network are realizable, but the 4th order network at (b) requires an 8.9 nH inductor, which is much too large to fabricate at 9.5 GHz. The only other 4th order topology that can potentially allow easy bias introduction (without adding a large RF choke) is shown at (c). The response of this network is essentially the same as that for the network at (b), but the shunt inductor in (c) (0.045 nH) is too small to allow direct introduction of gate bias. Therefore, and for different reasons, neither of the 4th order input topologies provide a practical monolithic solution to our matching problem.

Figure 2.30 compares the gain responses of the 3rd and 4th order input networks. While, in this case, the 4th order response is closer to an "ideal" sloped response than is that of the 3rd order network, it is not realizable monolithically, and the 3rd order response would be a satisfactory starting point for optimization in most applications.

The large signal output model for the device (at an output power level of +21 dBm) is shown in Figure 2.31, and the output matching networks are shown in Figure 2.32. Both networks were designed for a flat match across the band, and the ripple specifications are very similar.

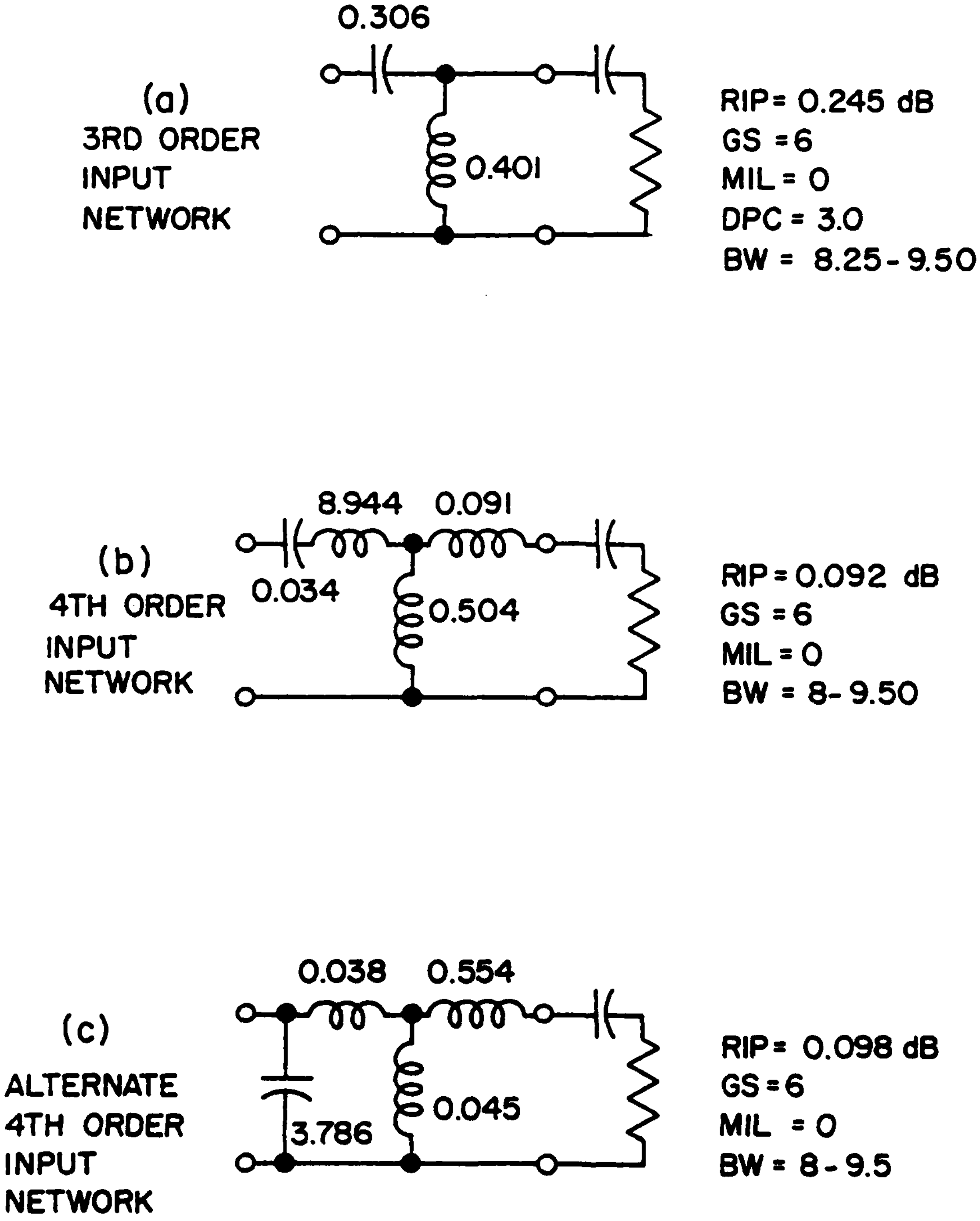


Figure 2.29. Power Amplifier Input Networks

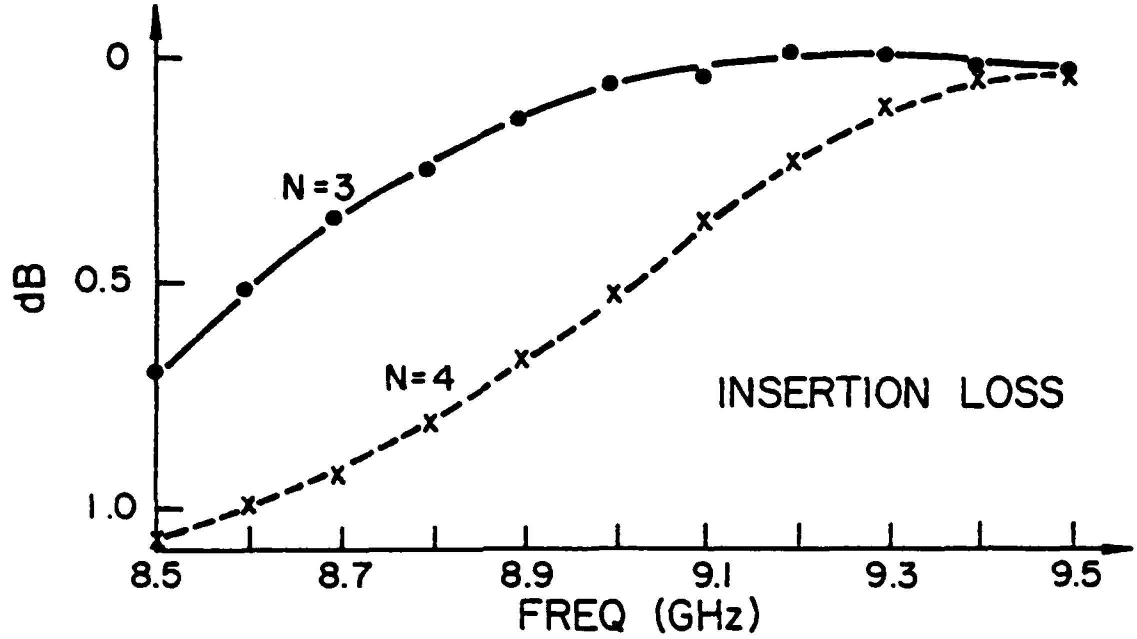


Figure 2.30. Power Amplifier Input Network Performance

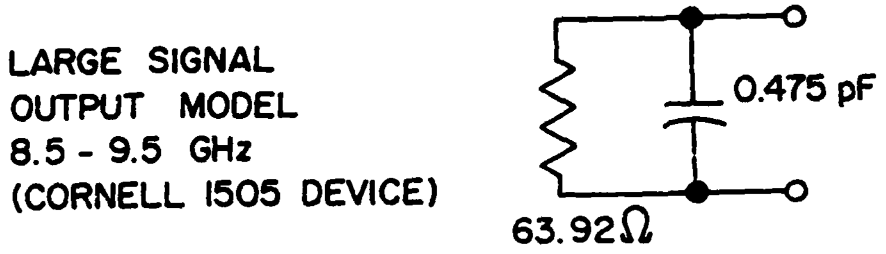


Figure 2.31. Power FET Output Model

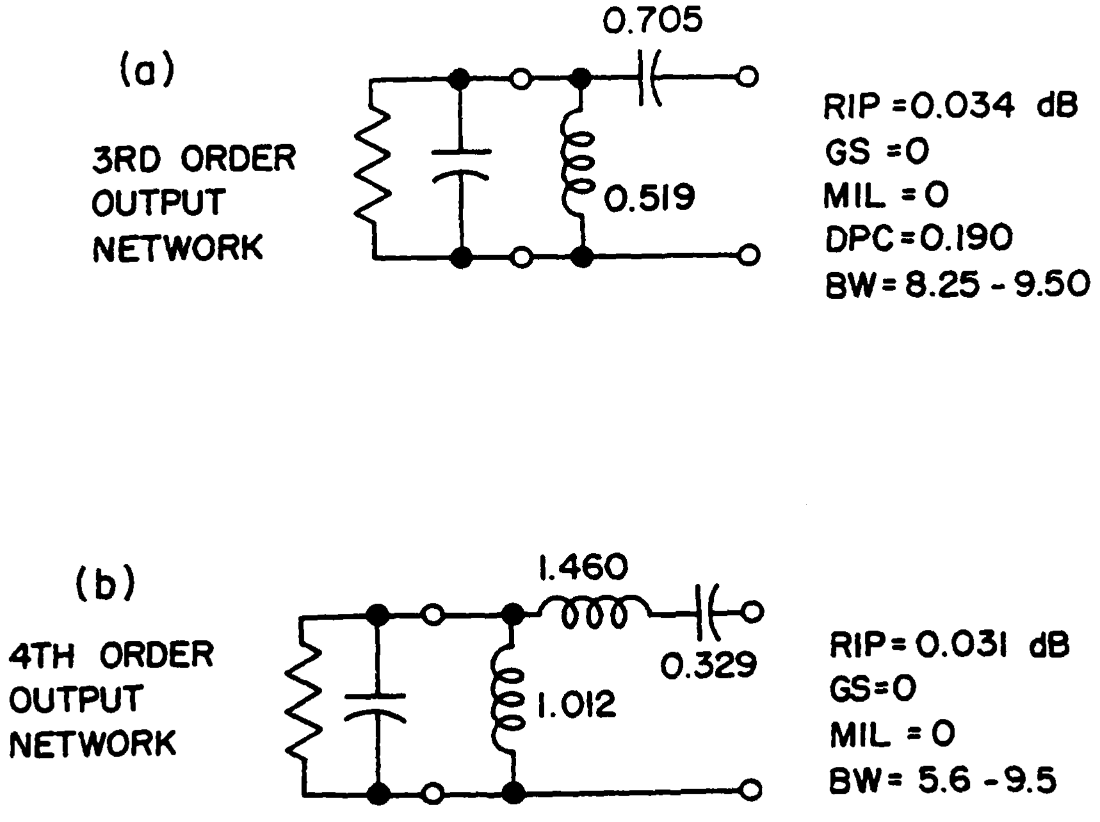


Figure 2.32. Power Amplifier Output Networks

No Norton Transformation was necessary for this 4th order network, so only 1 additional inductor is required.

Figure 2.33 shows the gain performance of both networks; there is no significant difference in the two responses (less than 0.03 dB, worst case, across the band).

2.13.3. Summary

Very often, a narrowband FET matching problem can be solved with a 3rd order network instead of with one of 4th order. Although this is not

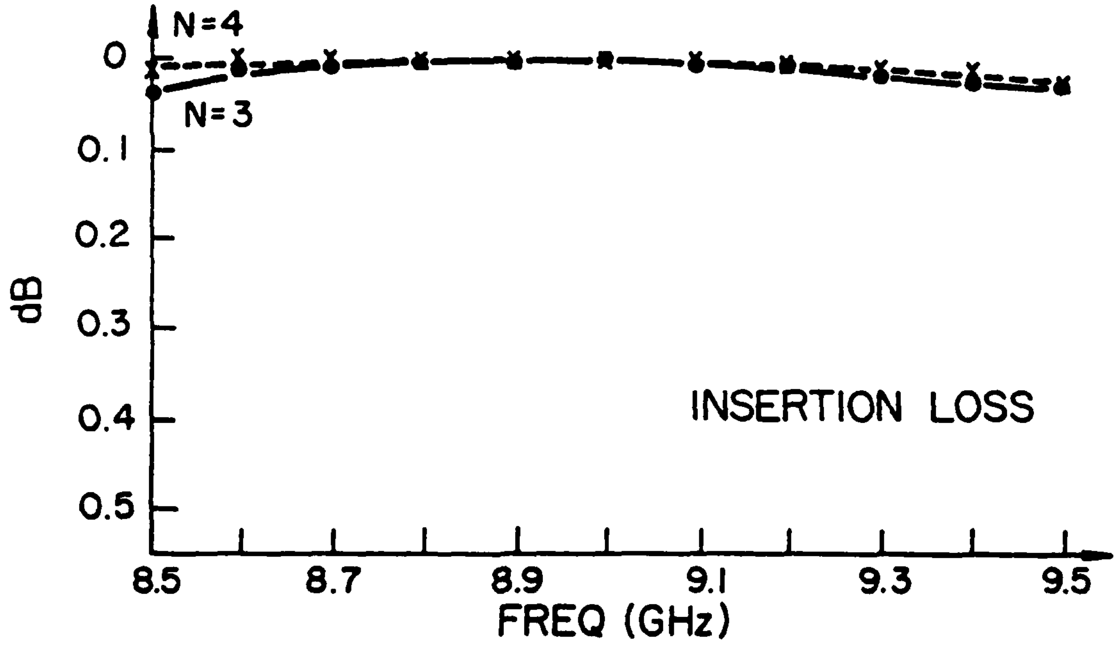


Figure 2.33. Power Amplifier Output Network Performance

always possible. the 3rd order networks, when usable, provide matching networks requiring substantially less chip area than corresponding 4th order networks with only slight degradation in circuit performance.

In practical monolithic GaAs amplifiers, the smaller size of the 3rd order network, when compared to a corresponding 4th order network, will almost always greatly overshadow any slight performance penalty incurred.

Chapter 3

Distributed Element Broadband Network Design (Arbitrary Order)

In some monolithic applications distributed element matching networks are more convenient to use than lumped element networks. For these instances it would be advantageous to be able to use odd-order networks, just as it is with lumped elements. Our goal is still to design broadband matching networks with the minimum number of elements, and once the theory has been extended to allow odd-order networks this will often mean 3rd order networks.

This chapter will describe the necessary modifications to the lumped element synthesis procedure to permit synthesis of odd- or even-order distributed element networks. These results represent an extension of the existing techniques which are limited to even-order.

3.1. Equiripple Gain Function Synthesis - Even or Odd Order

With a few exceptions, the overall network synthesis algorithm described in § 2.1 for lumped elements remains valid for distributed elements. Rather than conducting the gain function synthesis in the frequency domain (ω^2), however, for distributed element networks it is conducted in the Richards' Transformation domain (Ω^2) as discussed below. The other change is that the element extraction step is conducted differently in order to realize the network as shorted and open-circuited stubs and transmission line sections instead of as lumped inductors and capacitors.

The distributed element networks to be synthesized are assumed to consist of an arbitrary cascade of commensurate length unit elements (cascade transmission line sections) as well as series or shunt open-circuited or short-circuited stubs. The total number of circuit elements may be even or odd.

Gain function synthesis is carried out in the Richard's Transformation domain [9]

$$\Omega = \tan \omega T \tag{3-1}$$

where ω is frequency, and T is the delay length of the commensurate lines. The transfer function of the distributed element network can be conveniently expressed in the following form [10]

$$G(x) = \frac{Kx^k(1 + x)^q}{P_N(x)} \tag{3-2}$$

where: $x = \Omega^2$, the square of the Richards' Transformation variable

K = gain constant chosen to maintain realizability
$(0 \le G(x) \le 1 \quad \forall\ x > 0)$

k = number of high-pass elements in the network
(known once the network topology has been specified)

q = number of unit elements in the network
(known once the network topology has been specified)

$P_N(x)$ = an Nth order polynomial in x, whose coefficients are to be determined

This expression is slightly different from the G(x) form used with lumped elements (eq (2-1)), but a similar procedure will be used to form an equiripple approximation to a specified gain function by determining K and the coefficients of $P_N(x)$.

3.1.1. Approximation Algorithm

Since the general gain function approximation strategy for distributed elements is the same as that for lumped elements, the detailed explanation of the algorithm presented in § 2.4.2 will not be repeated here. This section will primarily summarize the modified equations used in the computations.

As in the lumped element case, the approximation is specified over N critical points when N is odd, and N+1 critical points when N is even. After choosing the initial estimate of the critical points (x_i), setting K=1, and computing the required gain levels at each critical point, we have

$$G_{req}(x_i) = \frac{x_i^k(1 + x_i)^q}{P_N(x_i)}, \qquad i = 1,2,\ldots,N \qquad (3\text{-}3)$$

$$P_N(x_i) = a_0 + \sum_{j=1}^{N} a_j * x_i^j$$

after re-arranging, we have

$$a_0 + \sum_{j=1}^{N} a_j * x_i^j = \frac{x_i^k(1 + x_i)^q}{G_{req}(x_i)}, \qquad i = 1,2,\ldots,N \qquad (3\text{-}4)$$

For odd N, we again choose a_0 as a parameter, and eq (3-4) then represents a system of N equations in N unknowns, since $G_{req}()$ is specified at N critical points. For even N, a_0 is also treated as a variable; the N+1 critical points specified allow determination of all N+1 coefficients of $P_N(x)$, and eq (3-4) represents N+1 equations in N+1 unknowns.

Equation (3-4) may be conveniently represented in matrix notation by

$$X \bar{a} = \bar{b} \tag{3-5}$$

where, <u>for odd N</u>

$$X = \begin{bmatrix} x_1^N & x_1^{N-1} & \cdot & x_1^2 & x_1 \\ x_2^N & x_2^{N-1} & \cdot & x_2^2 & x_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_N^N & x_N^{N-1} & \cdot & x_N^2 & x_N \end{bmatrix} \tag{3-6}$$

$$\bar{a} = \left[a_N, a_{N-1}, \ldots, a_2, a_1 \right]^T \tag{3-7}$$

and

$$\bar{b} = \left[\frac{x_1^k(1+x_1)^q}{G_{req}(x_1)} - a_0 , \frac{x_2^k(1+x_2)^q}{G_{req}(x_2)} - a_0 , \quad \ldots \quad , \frac{x_N^k(1+x_N)^q}{G_{req}(x_N)} - a_0 \right]^T \tag{3-8}$$

For *even N*, we have

$$X = \begin{bmatrix} x_1^{N+1} & x_2^{N} & \cdot & x_1 & 1 \\ x_2^{N+1} & x_2^{N} & \cdot & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_N^{N+1} & x_N^{N} & \cdot & x_n & 1 \\ x_{N+1}^{N+1} & x_{N+1}^{N} & \cdot & x_{N+1} & 1 \end{bmatrix} \qquad (3\text{-}9)$$

$$\bar{a} = \left[a_N, a_{N-1}, \ldots, a_2, a_1, a_0 \right]^T \qquad (3\text{-}10)$$

$$\bar{b} = \left[\frac{x_1^k(1+x_1)^q}{G_{req}(x_1)}, \frac{x_2^k(1+x_2)^q}{G_{req}(x_2)}, \ldots, \frac{x_N^k(1+x_N)^q}{G_{req}(x_N)}, \frac{x_{N+1}^k(1+x_{N+1})^q}{G_{req}(x_{N+1})} \right]^T \qquad (3\text{-}11)$$

We again solve eq (3-5) by LU factorization for best numerical stability.

This completes the first part of the approximation algorithm, and the coefficient vector, $\bar{a}$, is now available. The remainder of the procedure deals with determining the new critical points of the response via the Newton-Raphson Method, and this is accomplished in exactly the same manner as it is for lumped elements as described in § 2.4.2 (only the functions and derivatives evaluated during the procedure are different).

3.1.2. Implementation

The above algorithm has been implemented in a computer subroutine for solving the distributed element approximation problem with functions of arbitrary order and combined with other existing subroutines to form a complete distributed element network synthesis computer program (designated ADIST). A commented FORTRAN listing of the approximation subroutine (designated ODDIST) is presented in Appendix A.

3.2. Effects of Varying the Denominator Polynomial Constant (DPC)

The general remarks made in § 2.5 in reference to the realizability constraints on the choice of DPC in the lumped element case are also valid for the distributed element case. DPC must always be chosen to be positive, and there will be some maximum DPC value, dependent upon the particular gain function specifications in each case, which results in a realizable gain function. (The results presented in Chapter 4 show how this maximum realizable DPC value may be determined in advance for 3^{rd} order networks.)

The following two sections present some examples of the effects of DPC variation in actual distributed element networks.

3.2.1. Gain-Bandwidth Behavior with DPC

Odd-order distributed element networks exhibit gain-bandwidth (GBW) variations as the value of DPC is changed, just as described for the lumped element networks. The exact nature of this variation is very dependent upon the combination of gain function specifications for each particular case, so no general statement can be made as to the overall shape of the GBW vs DPC curves.

Figure 3.1(a) shows an example of GBW behavior for the low-pass constraint case, and Figure 3.1(b) shows an example of the results for the high-pass constraint case. Cases also exist (not shown) where GBW reaches a maximum at some DPC value internal to the realizable interval, just as with lumped elements; the two cases shown in Figure 3.1 reach maximum GBW at one or the other end of the realizable interval. Both examples in Figure 3.1 are 3rd order networks synthesized for octave bandwidth, 0.1 dB ripple, zero gain slope, and zero minimum insertion loss.

3.2.2. Terminating Impedance Behavior with DPC

Just as with lumped elements, changing DPC in an odd-order distributed element network also alters the required terminating resistance for the network. Figure 3.2 shows two examples of this behavior, for the same two circuits (and same gain specifications) used to generate the gain bandwidth curves in Figure 3.1. Once again, the only general statement which may be made regarding required terminating resistance as DPC is changed is that it varies; the exact shape of the curve is dependent upon the particular combination of gain specifications in each case.

3.3. Choosing DPC

The denominator polynomial constant is used as an additional synthesis parameter when designing odd-order distributed element matching

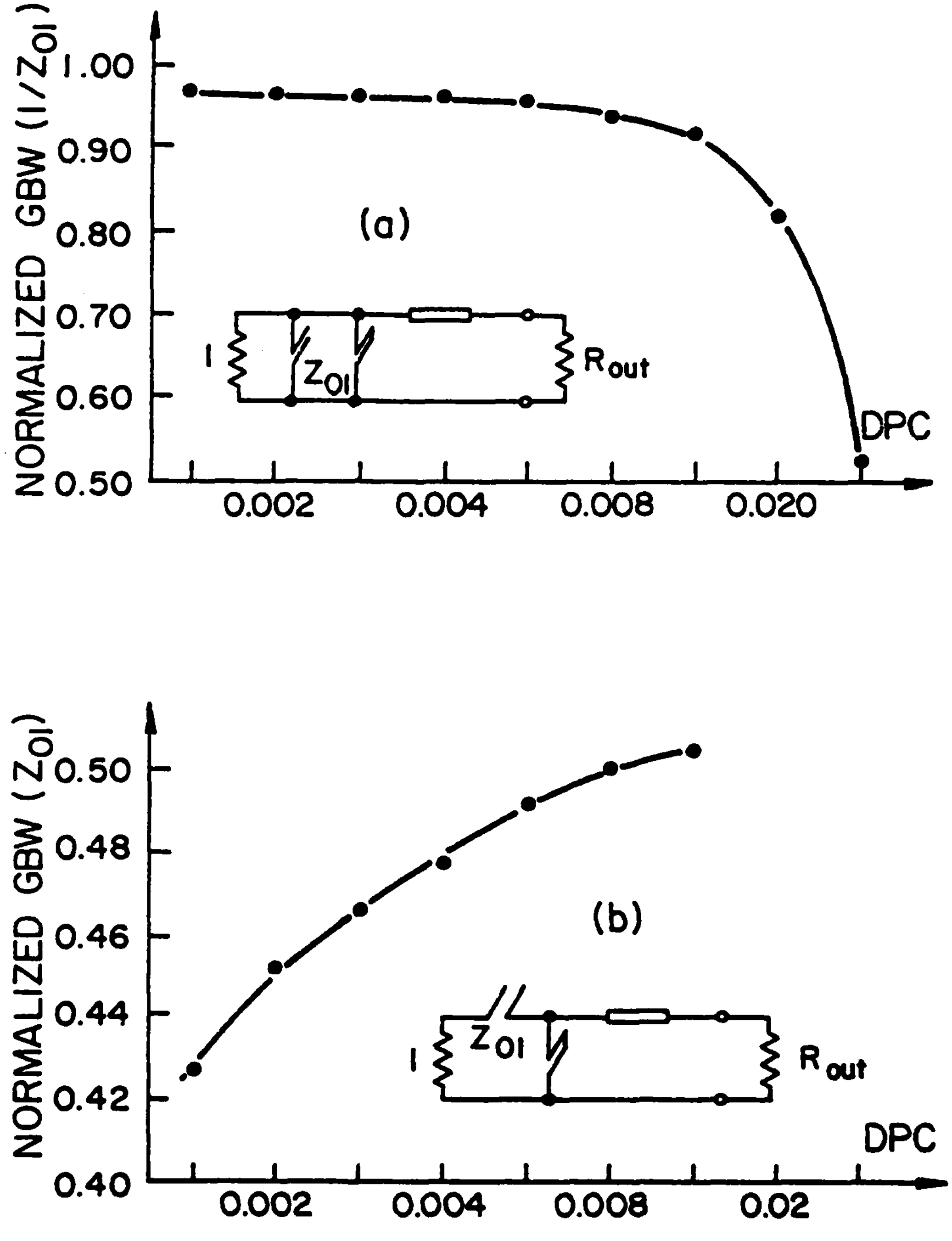


Figure 3.1. Gain-Bandwidth Effects of DPC Variation

networks, just as it is when designing lumped element networks. It can be chosen manually, along with the other synthesis parameters, in order to obtain a 3^{rd} or 5^{th} order distributed network that simultaneously absorbs the FET parasitic reactance and provides the needed impedance transformation, or it can be chosen by a computer search to maximize the GBW product of the network.

Such a computer search algorithm has been incorporated into the ADIST distributed element network synthesis program (the search strategy is identical to that described in § 2.6.2).

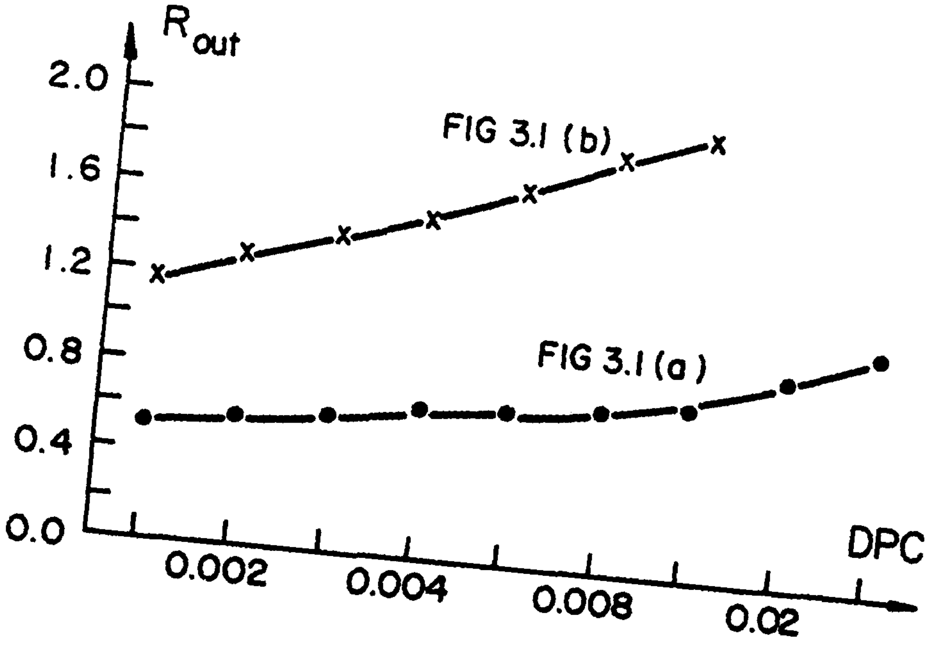


Figure 3.2. R_{out} Effects of DPC Variation

3.4. Practical Monolithic Input Networks

Usable monolithic topologies for distributed element networks are rather limited, since we can only allow shunt open- or short-circuited stubs and cascade transmission line sections. Series open- or short-circuited stubs cannot be realized in microstrip, and are therefore not included in any circuits intended for monolithic applications. (Series open-circuited stubs are included in some of the topologies discussed, but only to model the FET input reactance. These elements are not physically constructed.)

We require the circuits to include a shunt shorted stub; forming the "short" with a bypass capacitor will then produce a convenient point for bias introduction. Distributed synthesis does not allow including series capacitors, so a DC blocking capacitor will need to be added to all the example networks.

Figure 3.3 shows several example distributed input network topologies that can be realized monolithically. There are other possible topologies, as well, but these represent the simplest ones that fulfill our basic requirements. The series distributed FET input model is shown at the right of each circuit in the figure, and the assumed resistive source impedance for the amplifier is shown at the left.

The 3rd order network shown in Figure 3.3(a) is the simplest topology which is likely to be able to simultaneously absorb the FET parasitic while producing the correct impedance transformation. Figures 3.3(b) and (c) are 5th order networks, and the only difference between these two

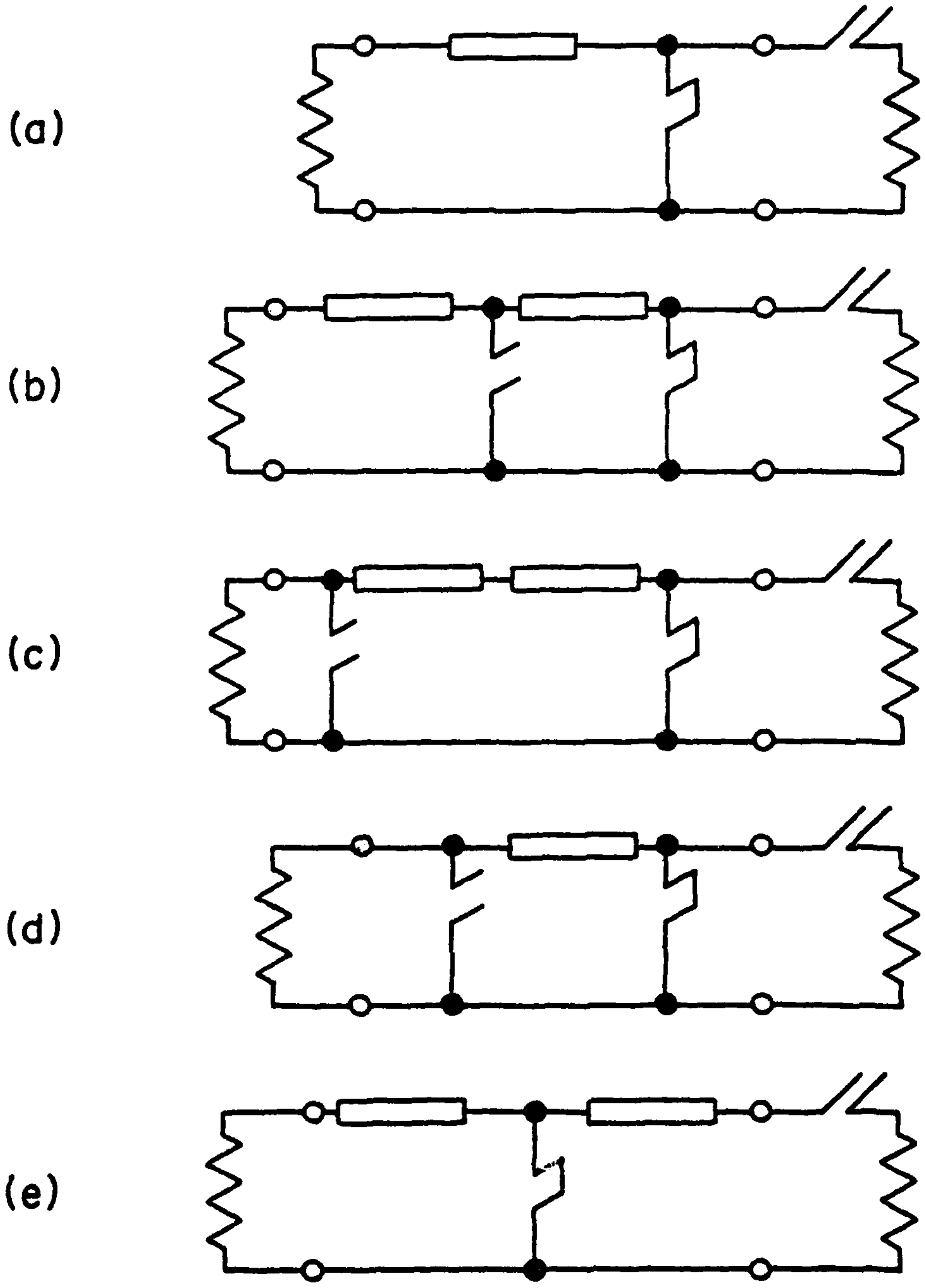


Figure 3.3. Practical Monolithic Input Networks (Distributed Elements)

for a fixed gain function specification will be the values of the required terminating impedance. Figures 3.3(d) and (e) are 4th order networks.

Several of the networks in Figure 3.3 also have the potential of allowing a Kuroda Transformation (discussed in § 3.8) to further adjust the terminating resistance.

3.5. Practical Monolithic Output Networks

Figure 3.4 shows several example distributed output network topologies that can be realized monolithically. The same basic circuit constraints discussed in § 3.4 also apply to these networks. The shunt distributed FET output model is shown at the left of each circuit in the figure, and the assumed resistive load impedance for the amplifiers is shown at the right. While there are other possible outpout topologies, the ones shown are the simplest that fulfill our basic requirements.

The circuit shown at Figure 3.4(a) is a 3rd order network, those at (b) and (c) are 4th order networks, and that at (d) is 5th order. All these circuits have the potential of allowing a Kuroda Transformation to adjust the terminating impedance.

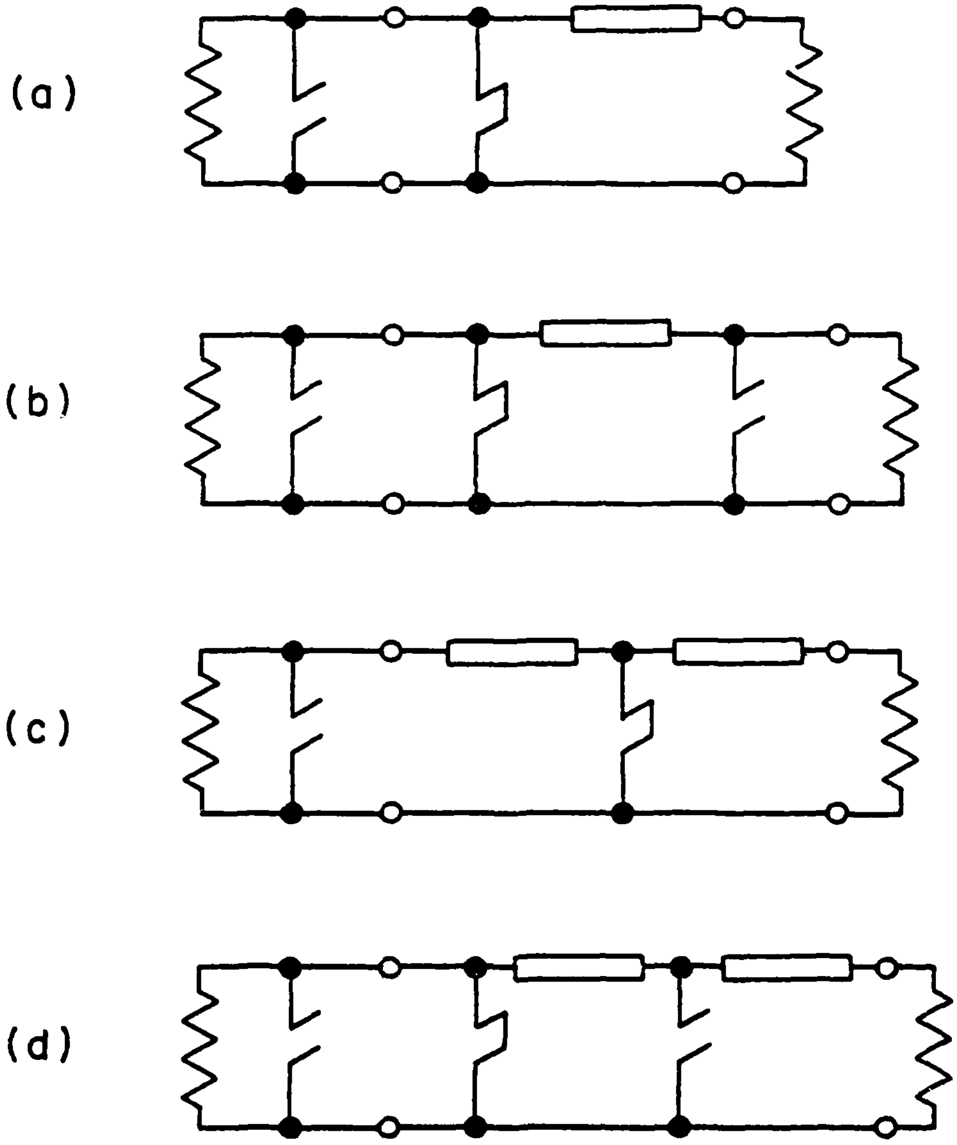


Figure 3.4. Practical Monolithic Output Networks (Distributed Elements)

3.6. Practical Monolithic Interstage Networks

Figure 3.5 shows several examples of distributed interstage network topologies that can be realized monolithically. The distributed FET output model is shown at the left of each circuit in the figure, and the distributed FET input model is shown at the right.

The circuits shown at Figure 3.5(a-c) are 4th and 5th order networks that may be directly synthesized. The first strategy to be used in designing circuits of this type is to attempt to find a combination of gain function parameters that simultaneously provide an impedance transformation from R_1 to R_2 while exactly absorbing Z_{01} and Z_{02}. If this cannot be done, it may be possible to solve the problem by absorbing Z_{01} exactly and then using a Kuroda Transformation to adjust the impedance transformation ratio. (Z_{02} will be properly absorbed in this process if the ratio of R_2 to Z_{02} prior to performing the transformation is the same as that required by the FET input model. Since R_2/Z_{02} is invariant under impedance scaling, once the output impedance of the network is adjusted to R_2 by the transformation, Z_{02} will also be scaled to the proper level for exact absorption.)

If neither of these strategies is successful, it is often possible to solve the interstage problem by cascading a 3rd or 4th order network at the output of the first FET with a 2nd order network at the input to the second FET. The resulting topologies are shown at Figure 3.5(d) and (e). The strategy is the same as with lumped elements; a 2nd order network is synthesized at the input to the second FET by adding shunt shorted stub Z^*

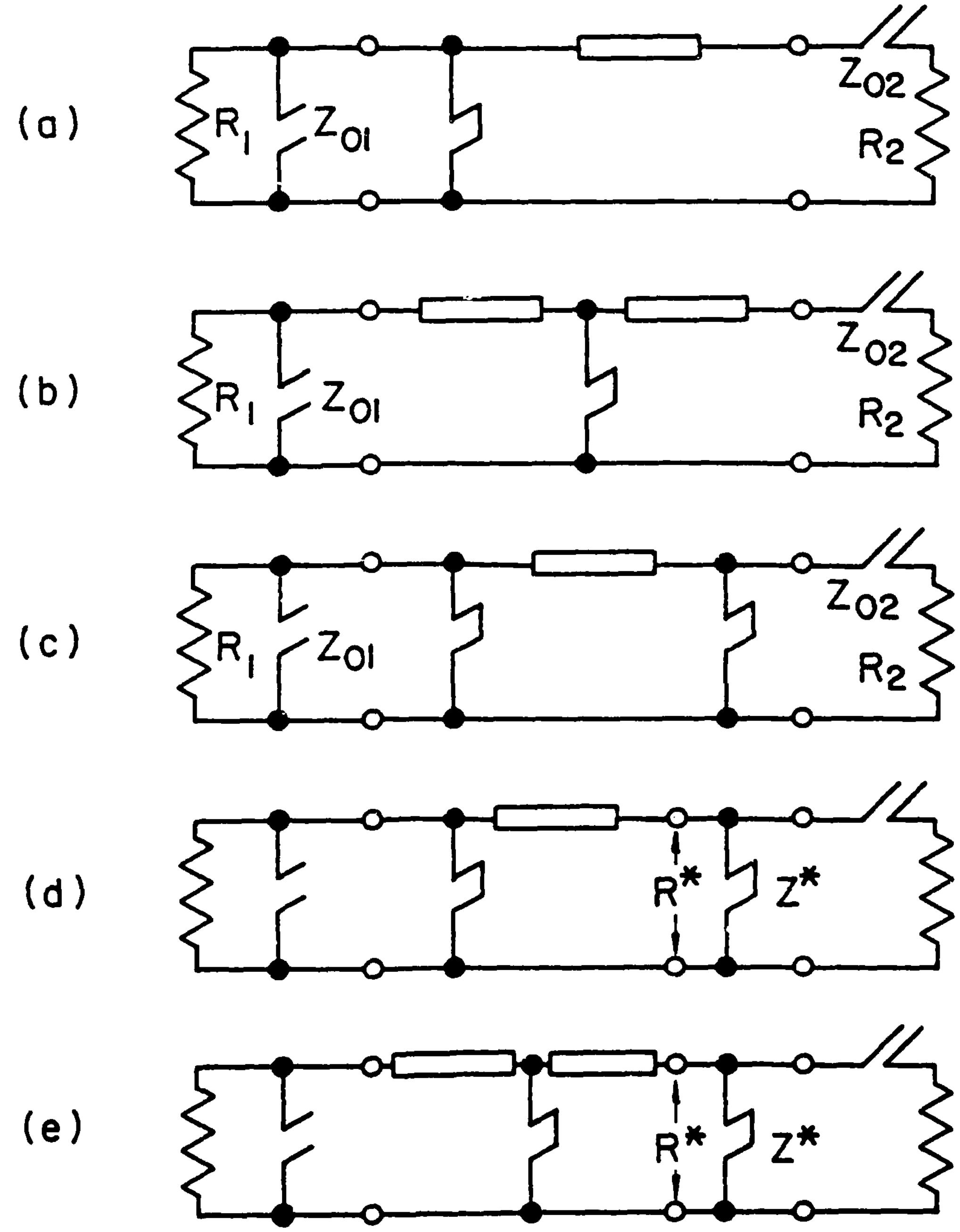
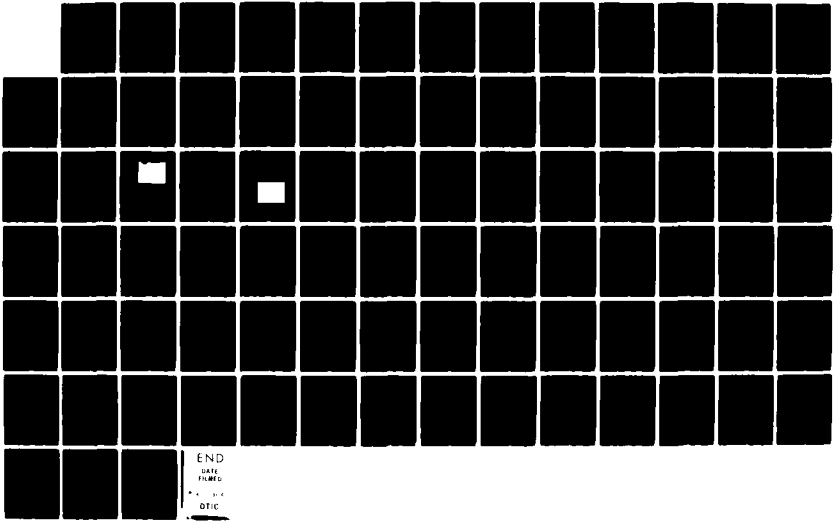


Figure 3.5. Practical Monolithic Interstage Networks (Distributed Elements)

AD-A132 474 COMPUTER-AIDED SYNTHESIS AND DESIGN OF MONOLITHIC MICROWAVE GAAS MESFET AMPLIFIERS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH J T DIJAK AUG 83

UNCLASSIFIED AFIT/CI/NR-83-19D F/G 9/5 NL

2/2

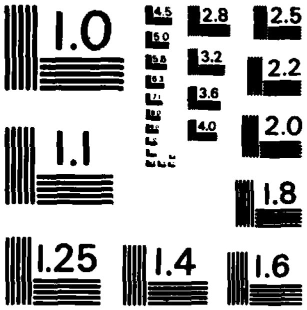


END

DATE FILMED

[illegible]

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

such that the FET parasitic is absorbed exactly. Then a 3rd order output network is synthesized for the first FET to match to the impedance level required at the input to the 2nd order network. Due to the loading and ripple effects between the two networks, it may be necessary to apply some computer optimization to the cascaded circuit to obtain a satisfactory overall gain response, especially for broadband networks.

As with lumped element interstage networks, it is again important that all possible distributions of the zeroes of the $s_{11}(p)$ function be examined during the search for a suitable interstage network.

3.7. Gain-Bandwidth Results

Odd-order distributed networks exhibit gain-bandwidth (GBW) characteristics similar to those of lumped element networks. The detailed discussion of § 2.11 will therefore not be repeated, but two example plots will be presented.

Figure 3.6 shows the normalized GBW product of 3rd, 4th, 5th, and 6th order distributed networks plotted against the ripple specification for the low-pass constraint case. The GBW performance of the 3rd and 5th order networks is much as we would expect compared to the 4th and 6th order networks, and is very similar to that shown in Figure 2.17 for lumped elements.

Figure 3.7 shows the GBW results for a set of distributed networks for the high-pass constraint case. Again, the results are very similar to the lumped element high-pass constraint results shown in Figure 2.18. The

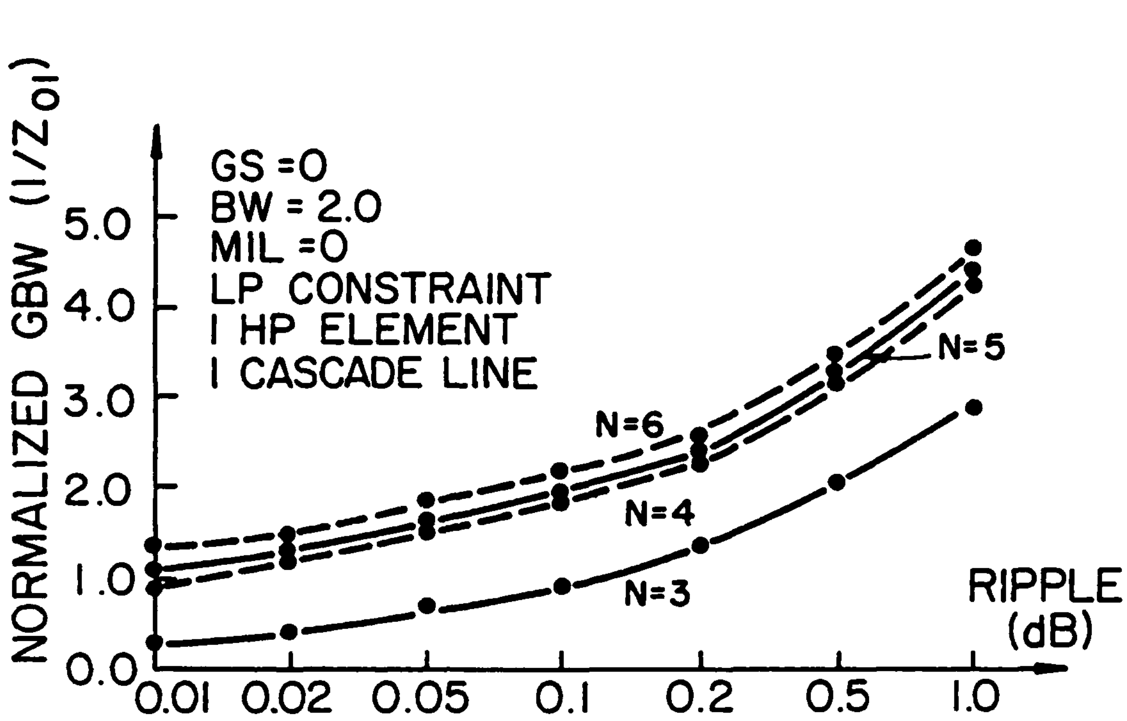


Figure 3.6 Gain-Bandwidth Results (LP, Distributed)

4th and 5th order GBW curves overlie each other, again because the root patterns of the $s_{11}(p)$ functions are so nearly identical. (The 6th order curve could not be extended to higher ripple values since it was impossible to generate a valid gain function that met those gain specifications with the specified topology.)

The odd-order results here for both the low-pass and high-pass constraint cases reflect the maximum GBW possible for each value of ripple, obtained by optimum choice of DPC.

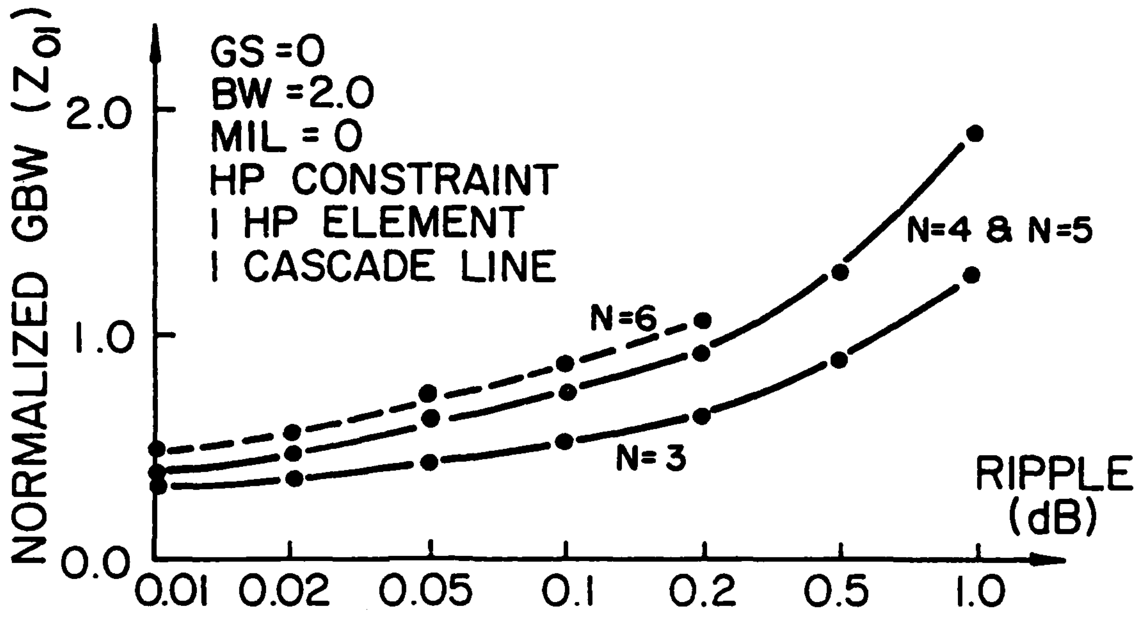


Figure 3.7. Gain-Bandwidth Results (HP, Distributed)

GBW results are not presented here for odd-order gain sloped distributed networks nor for the case of fixed ripple and varying minimum insertion loss. The distributed network curves are generally similar to the lumped element plots presented in Figures 2.19 through 2.22.

3.8. Kuroda Transformation

For those instances where a synthesized network cannot be found to exactly absorb an FET's parasitic reactance while providing a proper impedance transformation, it may be possible to employ a high-pass Kuroda

Transformation [11] to adjust the impedance transformation ratio to the required value.

Figure 3.8 illustrates the high-pass Kuroda Transformation. Figure 3.8(a) represents an initial distributed network with an unsatisfactory output impedance -- here it is drawn with an ideal transformer to bring the output impedance up to the desired R ohms. Since the network contains an adjacent shunt short-circuited stub and cascade line section, we may employ a Kuroda Transformation to eliminate the ideal transformer from the network.

The procedure is to "split" shunt shorted stub Z_1 into two parts (Z_{1a} and Z_{1b}) such that

$$N = 1 + \frac{Z_2}{Z_{1b}}$$

where N is the turns ratio of the ideal transformer to be eliminated from the network. Then by changing the topology to that shown in Figure 3.8(b) where

$$Z_A = \frac{Z_{1b} * Z_2}{Z_{1b} + Z_2}$$

and

$$Z_B = \frac{Z_{1b}^2}{Z_{1b} + Z_2}$$

the ideal transformer may be eliminated from the circuit.

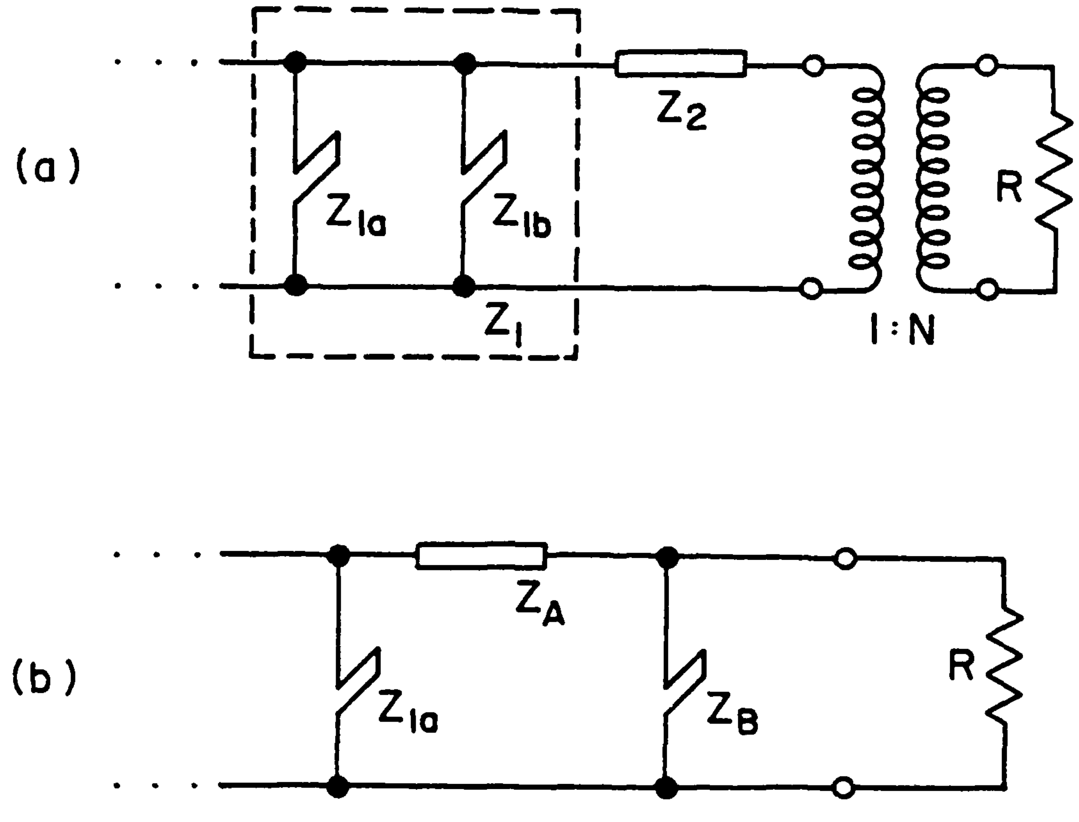


Figure 3.8. Kuroda High-Pass Transformation

Chapter 4

Analytical Determination of Maximum Denominator Polynomial Constant for Realizability

Chapter 2 discussed the limited range of the denominator polynomial constant (DPC) that can result in realizable network gain functions for odd-order networks. This realizable range is bounded by zero and some positive value dependent upon the particular gain function specifications for each case. It would be desirable to be able to determine, in advance, this positive limit for DPC so the circuit designer would have this information available at the time of synthesis.

This chapter presents techniques for analytically determining the maximum value for DPC that produces a realizable gain function for both lumped and distributed element 3^{rd} order networks with arbitrary topology, bandwidth, ripple, and gain slope. The procedure is completely general, and applies to all 3^{rd} order networks.

4.1. Lumped Elements

Figure 4.1 shows a generic 3^{rd} order bandpass gain function, of arbitrary bandwidth, ripple, and gain slope, that illustrates the mathematical problem being solved. The three critical frequencies of the response are designated x_1, x_2, and x_3; for the lumped element case we can simplify matters somewhat by normalizing the problem so that x_3=1. Let the

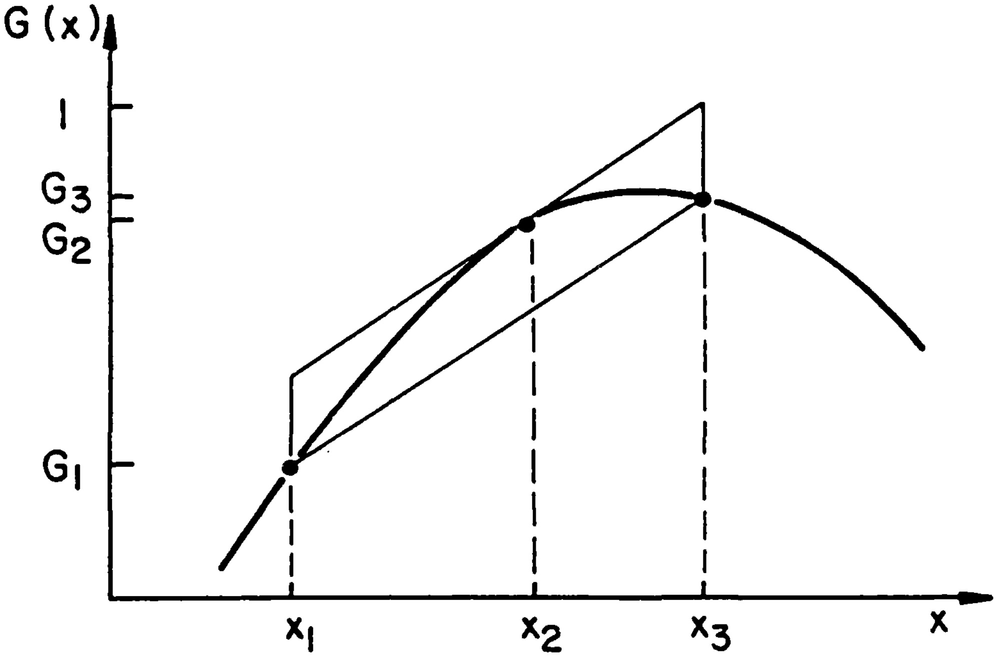


Figure 4.1 3^{rd} Order Gain Response

specified gain levels at these critical points be G_1, G_2, and G_3, respectively.

As in Chapter 2, the 3^{rd} order lumped-element bandpass gain function to be used is

$$G(x) = \frac{x^k}{Ax^3 + Bx^2 + Cx + D} \qquad (4\text{-}1)$$

where the numerator gain constant (K) has been set to 1 and k is the number of high-pass elements in the network. (k = 1 or 2 for the bandpass networks of interest.) We wish to set up a system of equations using eq (4-1) and the desired gain response specifications that allow us to determine the maximum value of D, above, that results in a positive coefficient for the high order denominator term (A).

Using the specified gain levels at the three critical points, we can immediately write the following three relations:

$$Ax_1^3 + Bx_1^2 + Cx_1 + D = \frac{x_1^k}{G_1} \quad (4\text{-}2)$$

$$Ax_2^3 + Bx_2^2 + Cx_2 + D = \frac{x_2^k}{G_2} \quad (4\text{-}3)$$

$$A + B + C + D = \frac{1}{G_3} \quad (4\text{-}4)$$

Where we have used the fact that $x_3=1$ for this case.

A fourth equation is provided by the constraint that the slope of the gain function at the center critical point be equal to the gain slope specified for the response. If we let S be this specified slope, then we require that $G'(x)\big|_{x=x_2} = S$. It can be shown that

$$G'(x) = \frac{(Ax^3+Bx^2+Cx+D)kx^{k-1} - x^k(3Ax^2+2Bx+C)}{(Ax^3 + Bx^2 + Cx + D)^2} \quad (4\text{-}5)$$

Evaluating at $x=x_2$, this relation can be simplified considerably by using eq (4-3)

$$G'(x)\bigg|_{x=x_2} = \frac{G_2}{x_2^k}(kx_2^{k-1}) - \frac{G_2^2}{x_2^k}(3Ax_2^2 + 2Bx_2 + C) \quad (4\text{-}6)$$

Setting eq (4-6) equal to our specified slope, S, and re-arranging, produces an equation which may be added to the three above to yield a solvable system.

$$3x_2^2A + 2x_2B + C = \frac{kx_2^{k-1}}{G_2} - \frac{Sx_2^k}{G_2^{\;2}} \qquad (4\text{-}7)$$

By using x_2 as a parameter to be chosen (instead of as a variable to be determined), we have a linear system of four equations in four unknowns (the coefficients A, B, C, D) which is solvable. This system can be conveniently represented in matrix notation by

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1^{\;3} & x_1^{\;2} & x_1 & 1 \\ x_2^{\;3} & x_2^{\;2} & x_2 & 1 \\ 3x_2^{\;2} & 2x_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \frac{1}{G_3} \\ \frac{x_1^k}{G_1} \\ \frac{x_2^k}{G_2} \\ \alpha \end{bmatrix} \qquad (4\text{-}8)$$

corresponding to

$$X \; \bar{a} = \bar{b} \qquad (4\text{-}9)$$

where

$$\alpha = \frac{kx_2^{k-1}}{G_2} - \frac{Sx_2^k}{G_2^2} \qquad (4\text{-}10)$$

Once x_2 is chosen, the matrix X is determined, and we can calculate the values for each element of $\bar{b}$ as soon as the gain function is specified, so we can solve for the coefficient vector, $\bar{a}$. This is not quite the way we will proceed, however, since we desire a different type of "solution".

By using determinants we can derive an analytical expression for the value of A. We can then find the value of x_2 that makes A = 0. We know that this x_2 will be a unique solution because the equiripple solution to the approximation problem is itself unique as shown by Petersen [1]. Deriving the analytical expression for D, and substituting the x_2 value found to produce a zero value for A will, in turn, show the corresponding D value that forces A to zero. This value for D (DPC) then represents the maximum value that produces a realizable gain function; any larger DPC value will produce a negative A value, which makes the gain function unrealizable.

Taking only the numerator of the full expression for A, and setting it equal to zero, we have:

$$
\begin{aligned}
0 = {} & \left[\alpha(x_1 - 1) + \frac{x_1^k}{G_1} - \frac{1}{G_3} \right] x_2^2 \\
& + \left[2\left[\frac{x_1}{G_3} + \frac{x_2^k(1 - x_1)}{G_2} - \frac{x_1^k}{G_1}\right] + \alpha(1 - x_1^2) \right] x_2 \qquad (4\text{-}11) \\
& + \left[\alpha(x_1^2 - x_1) + \frac{x_2^k(x_1^2 - 1)}{G_2} + \frac{x_1^k}{G_1} - \frac{x_1^2}{G_3} \right]
\end{aligned}
$$

where α is the function of S, x_2, and G_2 defined in eq (4-10).

Note that since we specify the reference gain function as an exponential function of x, the slope value, S, is also really an exponential function of x_2. In the general, arbitrary gain slope, case G_2 is also an exponential function of x_2. These complications make eq (4-11) extremely difficult, if not impossible, to solve using strictly analytical techniques, but not particularly difficult to solve via numerical equation solving techniques on a computer. So we use a computer algorithm to solve eq (4-11) for that value of x_2 $(x_1 < x_2 < 1)$ which produces $A = 0$.

The full expression for DPC is also quite complex. If we let

$$D = \frac{\mathrm{Num}(x_2)}{\mathrm{Den}(x_2)} \tag{4-12}$$

then

$$\begin{aligned}
\mathrm{Num}(x_2) = {} & \left[\frac{x_1}{G_3} - \frac{x_1^k}{G_1}\right] x_2^4 \\
& + \left[\alpha(x_1 - x_1^2) + 2\left[\frac{x_1^k}{G_1} - \frac{x_1^2}{G_3}\right]\right] x_2^3 \\
& + \left[\alpha(x_1^3 - x_1) + \frac{3x_2^k(x_1^2 - x_1)}{G_2} + \frac{x_1^3}{G_3} - \frac{x_1^k}{G_1}\right] x_2^2 \\
& + \left[\alpha(x_1^2 - x_1^3) + \frac{2x_2^k(x_1 - x_1^3)}{G_2}\right] x_2 \\
& + \frac{x_2^k(x_1^3 - x_1^2)}{G_2}
\end{aligned} \tag{4-13}$$

where α is again as defined in eq (4-10), and

$$\begin{aligned}
\mathrm{Den}(x_2) = {} & (x_1-1)x_2^4 + 2(1-x_1^2)x_2^3 + (x_1^3-3x_1^2+3x_1-1)x_2^2 \\
& + 2x_1(1-x_1^2)x_2 + x_1^2(x_1-1)
\end{aligned} \tag{4-14}$$

After substituting the known values into equations (4-13) and (4-14) the resulting maximum realizable value for D can be computed via eq (4-12).

This procedure has been coded into a set of computer subroutines (listed in Appendix A), and incorporated into the ARBSYN computer program where the maximum allowable DPC value for 3rd order networks is computed and presented to the operator at the appropriate time.

4.2. Distributed Elements

The approach to finding the maximum realizable DPC value for distributed element networks is the same as that for lumped elements, except that the equations are slightly different and more complicated. Figure 4.1 is also valid for the generic 3rd order gain response in the distributed case, with the important exception that we can no longer simply normalize so that $x_3=1$. (With distributed elements, due to the effects of chosen line length and the Richards Transformation, x_3 will often be some value *other* than 1.)

As in Chapter 3, the 3rd order distributed element bandpass gain function is

$$G(x) = \frac{x^k (1 + x)^q}{Ax^3 + Bx^2 + Cx + D} \qquad (4\text{-}15)$$

where k is the number of high-pass elements and q is the number of cascade lines in the network.

The three equations derived from the gain level specifications are

$$Ax_1^3 + Bx_1^2 + Cx_1 + D = \frac{x_1^k (1 + x_1)^q}{G_1} \qquad (4\text{-}16)$$

$$Ax_2^3 + Bx_2^2 + Cx_2 + D = \frac{x_2^k (1 + x_2)^q}{G_2} \qquad (4\text{-}17)$$

$$Ax_3^3 + bx_3^2 + Cx_3 + D = \frac{x_3^k (1 + x_3)^q}{G_3} \qquad (4\text{-}18)$$

The derivative expression evaluated at x_2 is

$$G'(x)\Big|_{x=x_2} = \frac{kG_2}{x_2} + \frac{qG_2}{(1+x_2)} - \frac{G_2^2}{x_2^k(1+x_2)^q}(3Ax_2^2 + 2Bx_2 + C) \qquad (4\text{-}19)$$

After setting eq (4-19) equal to the specified gain slope, S, and rearranging terms we have

$$3Ax_2^2 + 2Bx_2 + C = \frac{kx_2^{k-1}(1 + x_2)^q + x_2^k q(1 + x_2)^{q-1}}{G_2} - \frac{S}{G_2^2}\left[x_2^k(1 + x_2)^q \right] \qquad (4\text{-}20)$$

We again treat x_2 as a parameter, and the resulting system of linear equations can be described in matrix terms by

$$\begin{bmatrix} x_1{}^3 & x_1{}^2 & x_1 & 1 \\ x_2{}^3 & x_2{}^2 & x_2 & 1 \\ x_3{}^3 & x_3{}^2 & x_3 & 1 \\ 3x_2{}^2 & 2x_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \dfrac{x_1^k(1 + x_1)^q}{G_1} \\ \dfrac{x_2^k(1 + x_2)^q}{G_2} \\ \dfrac{x_3^k(1 + x_3)^q}{G_3} \\ \alpha \end{bmatrix} \qquad \textbf{(4-21)}$$

corresponding to

$$X \ \bar{a} \ = \ \bar{b} \qquad (4\text{-}22)$$

where

$$\alpha = \frac{kx_2^{k-1}(1 + x_2)^q + x_2^k q(1 + x_2)^{q-1}}{G_2} - \frac{S}{G_2^2}\left[x_2^k(1 + x_2)^q \right] \qquad (4\text{-}23)$$

We again derive an expression for A in terms of x_2 by using determinants, and solve for the value of x_2 that results in A = 0. This equation is

$$0 = \left[\alpha(x_1 - x_3) + \frac{x_1^k(1 + x_1)^q}{G_1} - \frac{x_3^k(1 + x_3)^q}{G_3}\right] x_2^2 \qquad (4\text{-}24)$$

$$+ \left[\alpha(x_3^2-x_1^2) + 2\left[\frac{x_1x_3^k(1+x_3)^q}{G_3} + \frac{(x_3-x_1)x_2^k(1+x_2)^q}{G_2} - \frac{x_3x_1^k(1+x_1)^q}{G_1} \right] \right] x_2$$

$$+ \left[\alpha(x_1^2x_3-x_1x_3^2) + \frac{(x_1^2-x_3^2)x_2^k(1+x_2)^q}{G_2} - \frac{x_1^2x_3^k(1+x_3)^q}{G_3} + \frac{x_3^2x_1^k(1+x_1)^q}{G_1} \right]$$

where α is as defined in eq (4-23).

Equation (4-24) is solved numerically, and the resulting value of x_2 is used to determine the corresponding maximum realizable value of D. Again using

$$D = \frac{\mathrm{Num}(x_2)}{\mathrm{Den}(x_2)} \tag{4-25}$$

we have, for the distributed case

$$\begin{aligned}
\mathrm{Num}(x_2) = {} & \left[\frac{x_1 x_3^k (1 + x_3)^q}{G_3} - \frac{x_3 x_1^k (1 + x_1)^q}{G_1} \right] x_2^4 \qquad (4\text{-}26) \\
& + \left[\alpha(x_1 x_3^2 - x_1^2 x_3) + \frac{2x_3^2 x_1^k (1 + x_1)^q}{G_1} - \frac{2x_1^2 x_3^k (1 + x_3)^q}{G_3} \right] x_2^3 \\
& + \left[\alpha(x_1^3 x_3 - x_1 x_3^3) + \frac{3(x_1^2 x_3 - x_1 x_3^2) x_2^k (1+x_2)^q}{G_2} + \frac{x_1^3 x_3^k (1+x_3)^q}{G_3} - \frac{x_3^3 x_1^k (1+x_1)^q}{G_1} \right] x_2^2 \\
& + \left[\alpha(x_1^2 x_3^3 - x_1^3 x_3^2) + \frac{2(x_1 x_3^3 - 2x_1^3 x_3) x_2^k (1 + x_2)^q}{G_2} \right] x_2 \\
& + \left[\frac{(x_1^3 x_3^2 - x_1^2 x_3^3) x_2^k (1 + x_2)^q}{G_2} \right]
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Den}(x_2) = {} & (x_1 - x_3) x_2^4 + 2(x_3^2 - x_1^2) x_2^3 + (x_1^3 - 3x_1^2 x_3 - 3x_1 x_3^2 - x_3^3) x_2^2 \qquad (4\text{-}26) \\
& + 2(x_1 x_3^3 - x_1^3 x_3) x_2 + (x_1^3 x_3^2 - x_1^2 x_3^3)
\end{aligned}$$

After substituting the known values into equations (4-26) and (4-27) the resulting maximum realizable value for D can be calculated via eq (4-25).

This procedure has also been coded into a set of computer subroutines (listed in Appendix A), and incorporated into the ADIST computer program for distributed network synthesis where the maximum allowable DPC value for 3rd order networks is calculated and presented to the operator at the appropriate time.

Chapter 5

Restricted Topology Matching Networks for Ease of Fabrication

When experimenting with different structures and doping profiles to optimize FET's intended for monolithic applications, it is convenient to have impedance matching topologies available that require no capacitors and only one level of metallization. These circuits would make it possible to quickly and easily fabricate single-stage monolithic amplifiers to realistically evaluate FET performance. Since these test amplifiers will require off-chip biasing (usually through bias tees), their use will be largely limited to the laboratory -- but there they can be very useful for demonstrating the capabilities of a particular FET design in a realistic amplifier environment.

5.1. Allowable Topologies

The circuit topologies which meet the restrictions described above are composed only of shunt open-circuited stubs and cascaded transmission line sections. Matching networks composed only of these two circuit elements require only one level of metallization, no RF shorts to the chip ground plane, and no capacitors. These traits make this class of circuit very desirable for some applications, but there are also disadvantages.

In addition to the problem of bias introduction, these circuits allow only indirect control of the impedance transformation ratio provided by the network. Since shunt short-circuited stubs are not allowed in these topologies, the high-pass Kuroda Transformation (§ 3.8) cannot be used to adjust impedance transformation ratio. As a result, the task of simultaneously providing a specific impedance transformation while exactly absorbing the FET parasitic reactance is much more difficult with these networks. Indeed, in many cases no directly synthesized network will be found that can simultaneously absorb the FET parasitic and match impedances. In these cases, however, using a cascade of two networks will usually solve the problem, though sometimes requiring computer optimization for broadband applications.

5.1.1. Input Networks

Several input network topologies that conform to these topology restrictions are shown in Figure 5.1. The 3rd order circuit shown at Figure 5.1(a) may be synthesized via the ADIST computer program for broadband applications, or by Smith Chart for narrowband applications. The 4th and 5th order topologies shown at Figure 5.1(b-d) can be directly synthesized via ADIST for either wide or narrowband applications, while the circuit at Figure 5.1(e) is formed by cascading 4th and 2nd order sections.

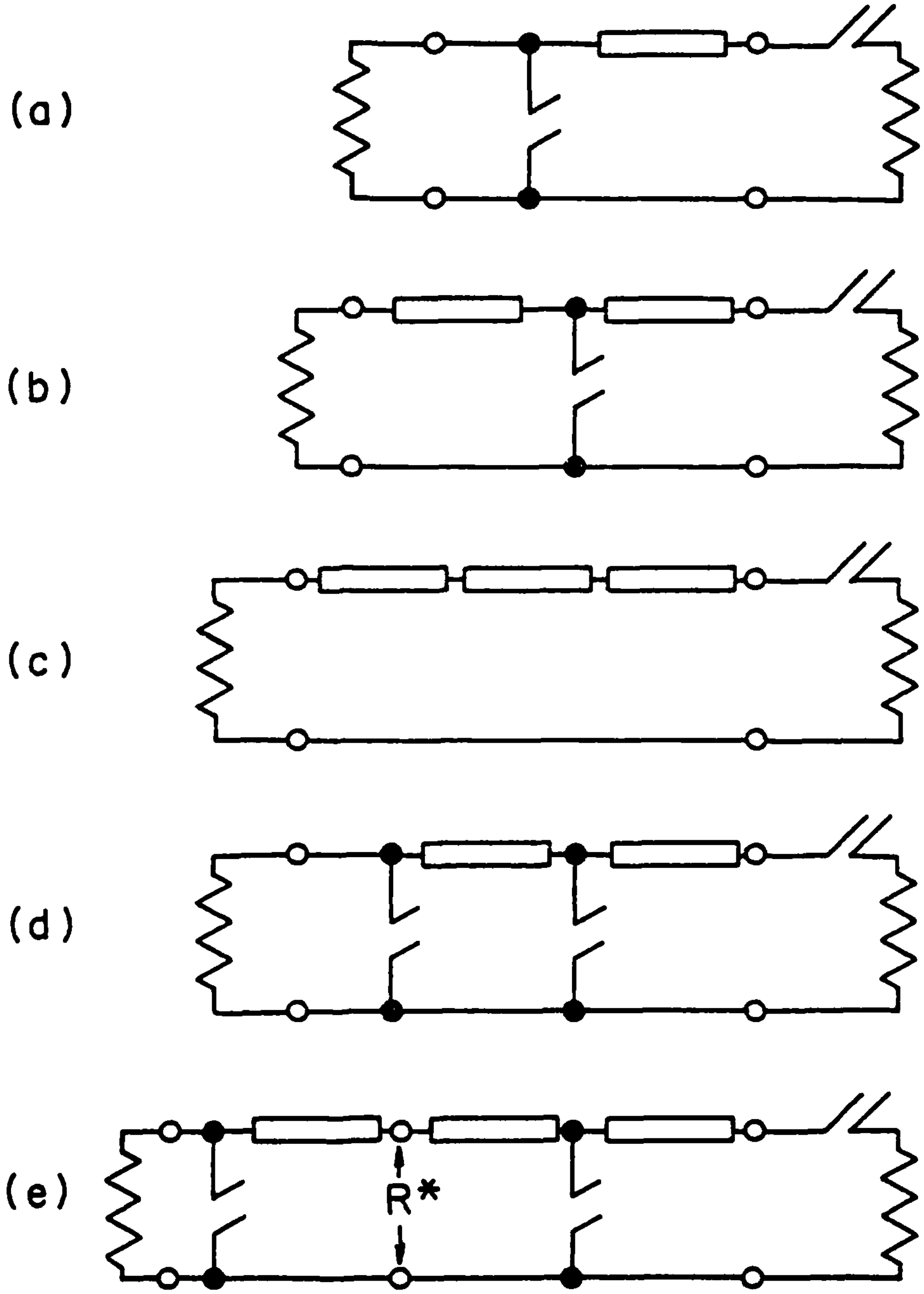


Figure 5.1. Restricted Topology Input Networks

The cascading procedure here is similar to that described in § 2.9 and § 3.6; this technique is even more important when no directly synthesized network can be found that simultaneously absorbs the device parasitic and produces the necessary impedance transformation.

5.1.2. Output Networks

Usable output topologies composed only of shunt open stubs and cascade lines are shown in Figure 5.2. The topology shown at Figure 5.2(a) may be synthesized by Smith Chart for narrowband applications, and those at (b) and (c) may be synthesized by the ADIST program for either narrow or wide bandwidths. The topology shown in Figure 5.2(d) is formed by cascading.

5.2. Single-Frequency Design Method

The simple networks of Figures 5.1(a) and 5.2(a) may be designed for a match at a single-frequency by using the Smith Chart. Figure 5.3 graphically illustrates the procedure.

Starting from the normalized FET input or output impedance (point A), a cascade transmission line section is used to proceed to point B. The length of this line is chosen so that after reflecting point B into the admittance plane (at point C), C lines on the g=1 circle. The match is then completed by moving from C to D by adding a shunt open-circuited stub.

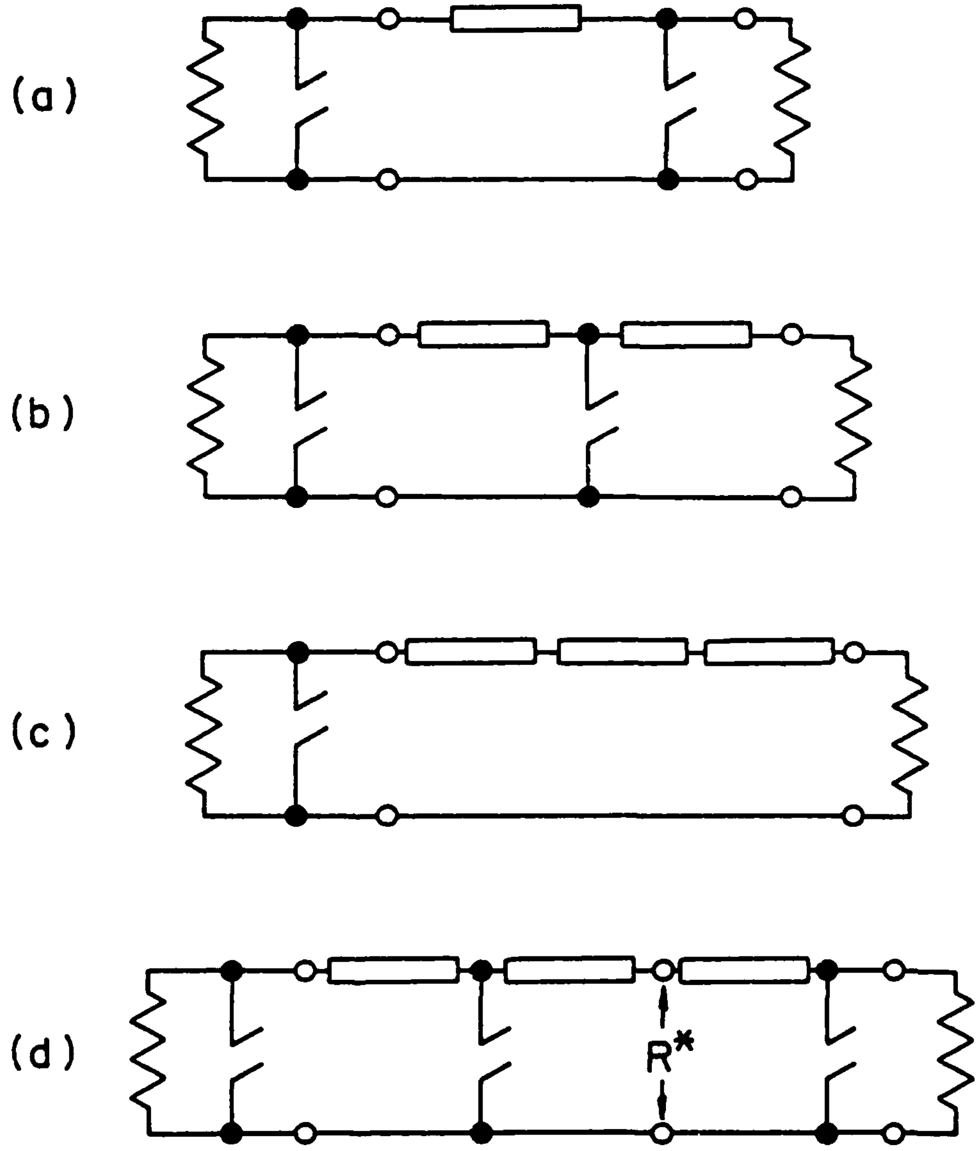


Figure 5.2. Restricted Topology Output Networks

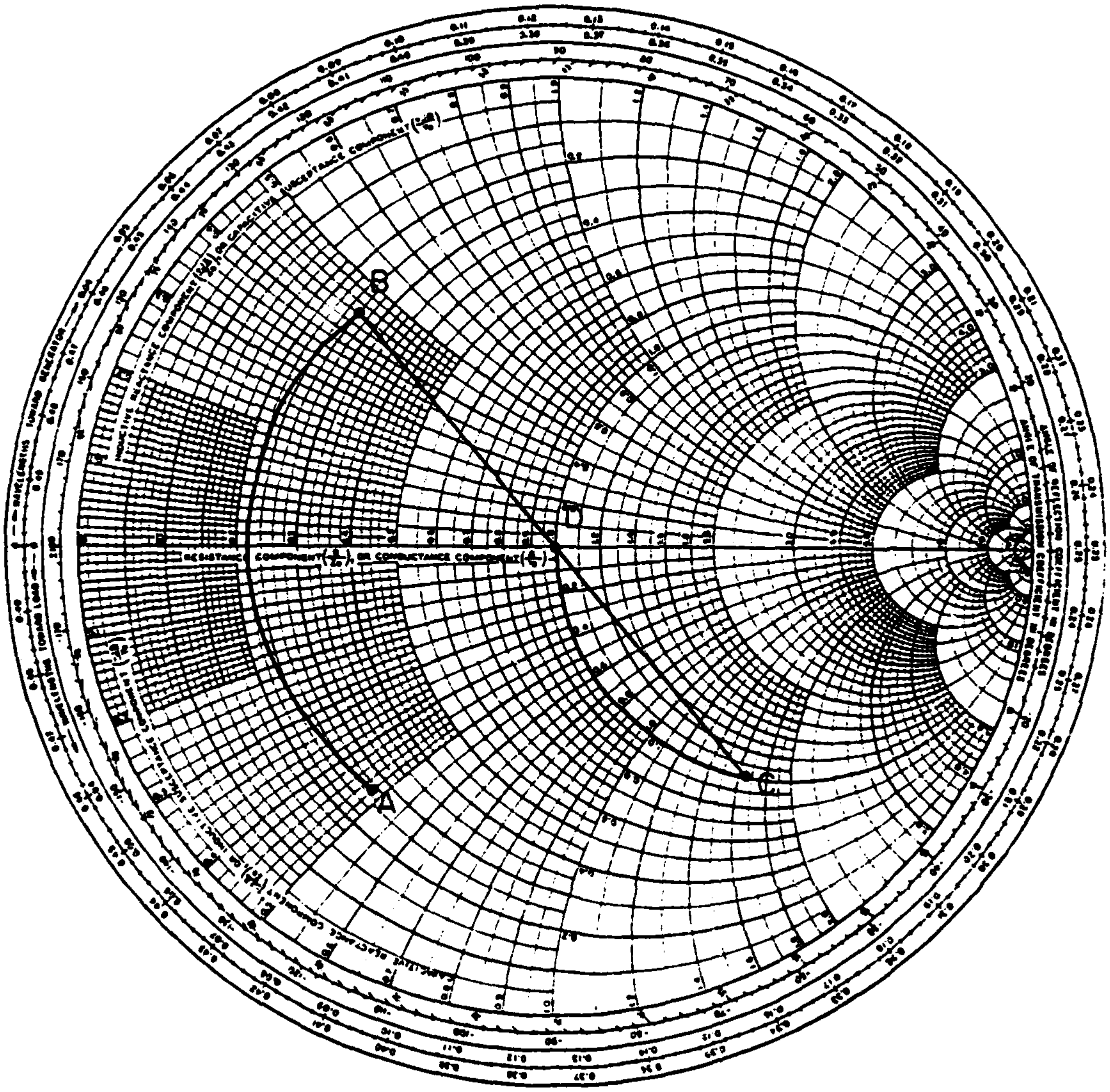

B
D
A
C
RESISTANCE COMPONENT (R/Zo), OR CONDUCTANCE COMPONENT (G/Yo)

Figure 5.3. Restricted Topology Single-Frequency Solution

5.3. Broadband Design Method

5.3.1. Direct Synthesis

It is always best to use a directly synthesized broadband network (rather than a cascaded design) when a suitable combination of topology and gain function specifications can be found to simultaneously absorb the FET's parasitic capacitance and provide the necessary impedance transformation. With the restricted topology networks of this section it is especially important that all possible topologies, bandwidths, ripple specifications, and $s_{11}(p)$ zero distributions be explored when seeking a solution, since Kuroda Transformations may not be used.

Due to this lack of flexibility, in many cases it will not be possible to find a suitable directly synthesized network. In these cases, a cascaded design may be used.

5.3.2. Cascaded Synthesis

5.3.2.1. Flat Match

The cascading procedure when an impedance match with no gain slope compensation is desired is as follows:

(1) Synthesize a zero gain slope 3rd or 4th order restricted topology network that meets the gain and bandwidth requirements while exactly absorbing the FET parasitic input (or output) capacitance -- but without regard to the required network terminating impedance (R^*).

(2) Synthesize a zero gain slope 2nd order restricted topology network that, over the specified band, produces an impedance transformation from R^* to the desired level at the source (or load) for the amplifier.

(3) Cascade these two circuits, analyze the resulting gain response, and apply computer optimization, if required, to yield a satisfactory gain response.

This procedure yields circuits like those shown in Figures 5.1(e) and 5.2(d). Very satisfactory results have been obtained with this technique up to octave bandwidth.

5.3.2.2. Gain Sloped Match

The cascading procedure for an impedance match with gain slope compensation is slightly different than that described above, and it is not possible to exactly synthesize a specified value of gain slope compensation. Due to mismatch effects between the two cascaded networks used, the final slope of the cascade is impossible to predict exactly. An approximate procedure is therefore used, and one or two iterations of the procedure may be required to obtain the desired slope in the final network. The procedure is as follows:

(1) Synthesize a 3rd or 4th order restricted topology network that provides one-half the desired gain slope compensation over the required band while exactly absorbing the FET parasitic capacitance -- again without regard to the required network terminating impedance (R^*).

(2) Synthesize a zero gain slope 2nd order restricted topology network that over twice the specified band produces an impedance transformation from R* to the desired level. (This network is designed over twice the specified band, with its center frequency at the upper edge of the specified passband. In this way, an exact impedance match is provided at the upper passband edge with increasing insertion loss at lower frequencies.)

(3) Cascade these two circuits and apply computer optimization to obtain a satisfactory gain slope.

(4) If necessary, increase or decrease the starting value of gain slope used in step (1) and repeat the procedure to obtain a more satisfactory result.

5.4. Restricted Topology Design Examples

Narrow and broadband monolithic FET power amplifiers were designed by using the above procedures, and the resulting amplifiers were fabricated on GaAs and tested by another student. Details of the device characterization, GaAs fabrication, and RF measurement procedures used can be found in [12].

5.4.1. Octave Band Amplifier

Figure 5.4 shows the 6-12 GHz monolithic power amplifier designed by using these techniques. The power FET was a 600 μm gate width design with a 0.8 μm gate length (additional details on the device can also be found in [12]). The figure shows the FET modelled unilaterally for design of the matching networks. The input model was obtained by optimization to the measured small-signal s_{11} characteristics of the device, while the output model was optimized to model the impedance that maximized output power from the device, as measured under large-signal conditions.

The input network shown in Figure 5.4 was formed by producing a single-frequency match at 12 GHz, the upper band edge. Analysis showed that this circuit yielded a smoothly decreasing gain response with 7.5 dB

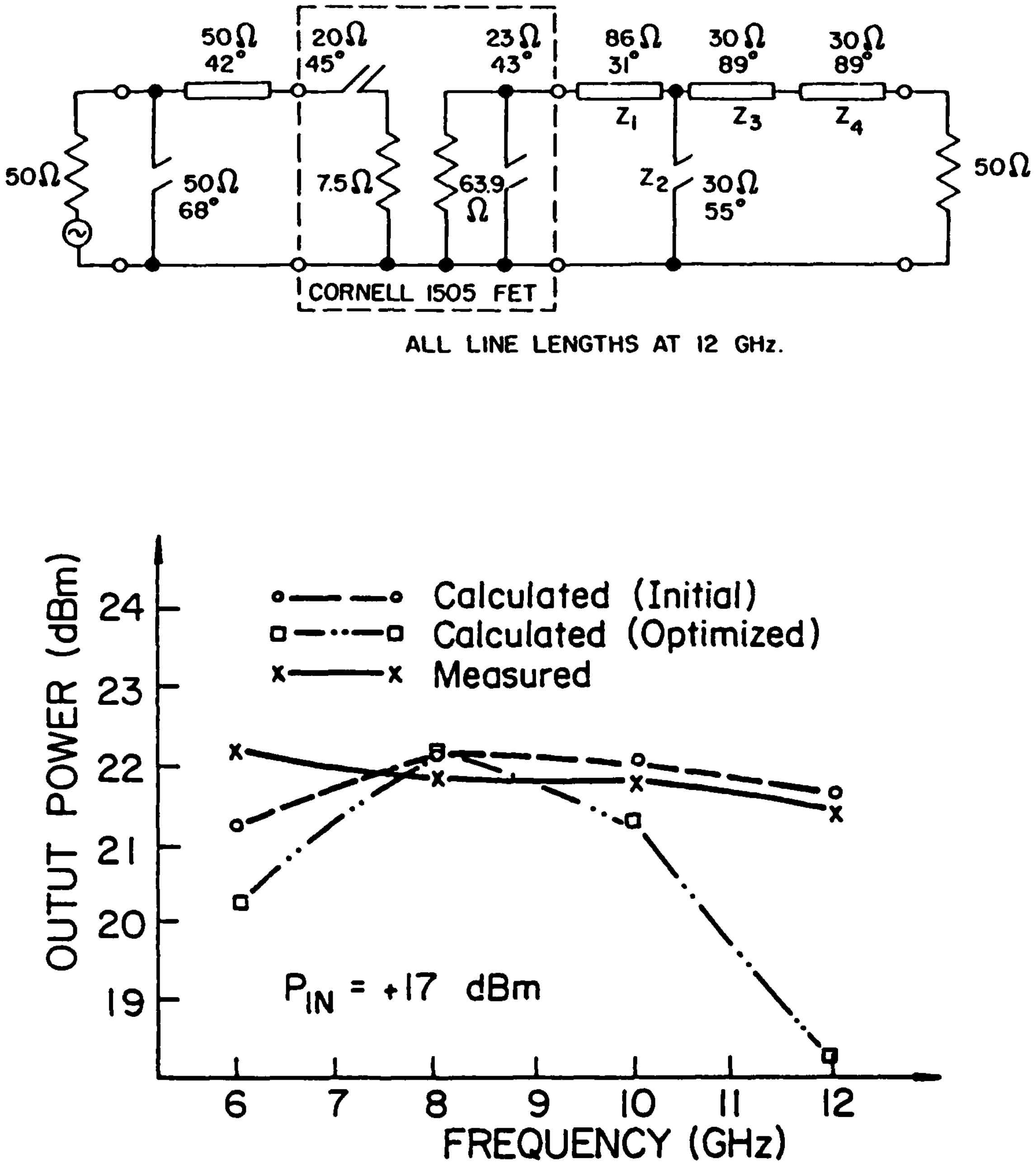


Figure 5.4. Restricted Topology Monolithic 6-12 GHz Power Amplifier

of insertion loss at the lower band edge, 6 GHz -- very satisfactory for gain slope compensation of this device. No computer optimization of this network was required.

The output network was formed by cascading 4th and 2nd order sections; it was designed for zero gain slope. (No directly synthesized network meeting the topology restrictions of this section could be found to exactly match the device to a 50 Ohm load over the 6 - 12 GHz band.) The FET's parasitic output capacitance was absorbed by a 4th order network (Z_1, Z_2, and Z_3 in conjunction with the FET's capacitance); this network required a 5.2 Ohm terminating impedance.

A 2nd order network consisting of the line section Z_4 and an open shunt stub (not shown) was then synthesized to accomplish the 5.2 to 50 Ohm impedance transformation over 6 - 12 GHz. The two networks were cascaded and computer optimization was applied to minimize insertion loss across the band. This optimization reduced the open shunt stub to an insignificant line length; the stub was then removed from the design and a final optimization performed.

The predicted and measured performance of the final amplifier circuit is also shown in Figure 5.4. The amplifier achieved 5 ± 0.5 dB power gain over 6 - 12 GHz when operating at the design drive level of +17 dBm.

Figure 5.5 shows the gain performance of the cascaded output network before and after computer optimization.

Figure 5.6 is a scanning electron microscope (SEM) photograph of the 6 - 12 GHz monolithic amplifier chip prior to scribing. (The distortion apparent in the image is due to irregularities in the electron beam scan at the extreme angle required to image this particular chip on the wafer.)

5.4.2. 10% Band Amplifier

Figure 5.7 shows the 8.5 - 9.5 GHz narrowband monolithic power amplifier designed by these techniques. The FET was the same as that used

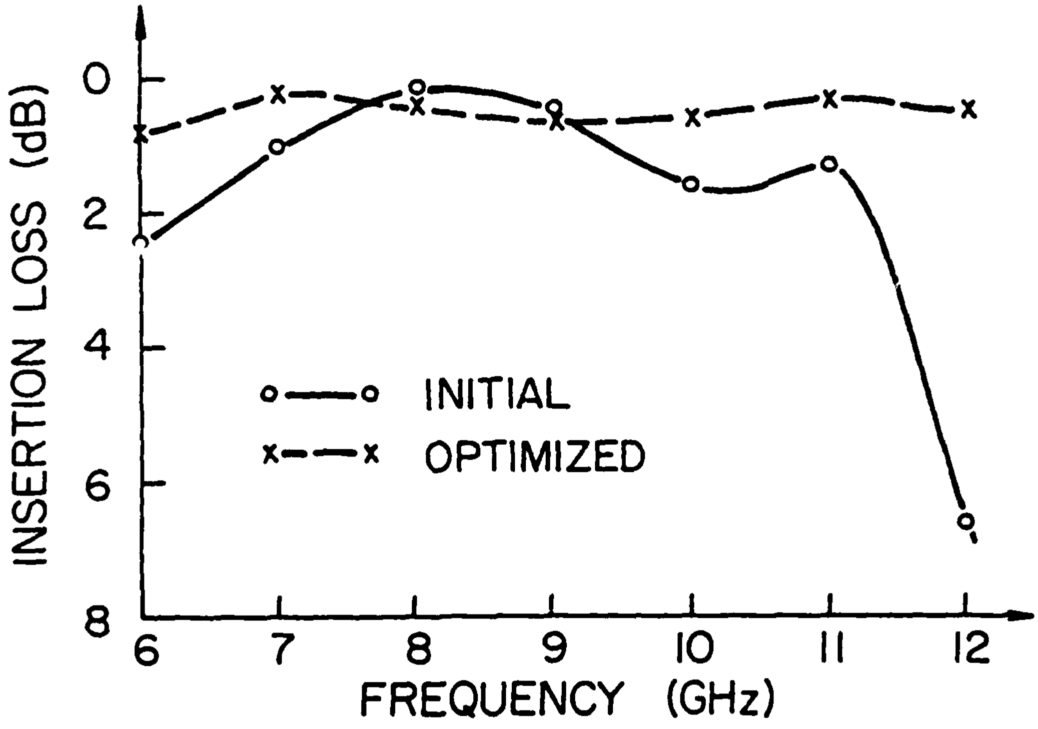


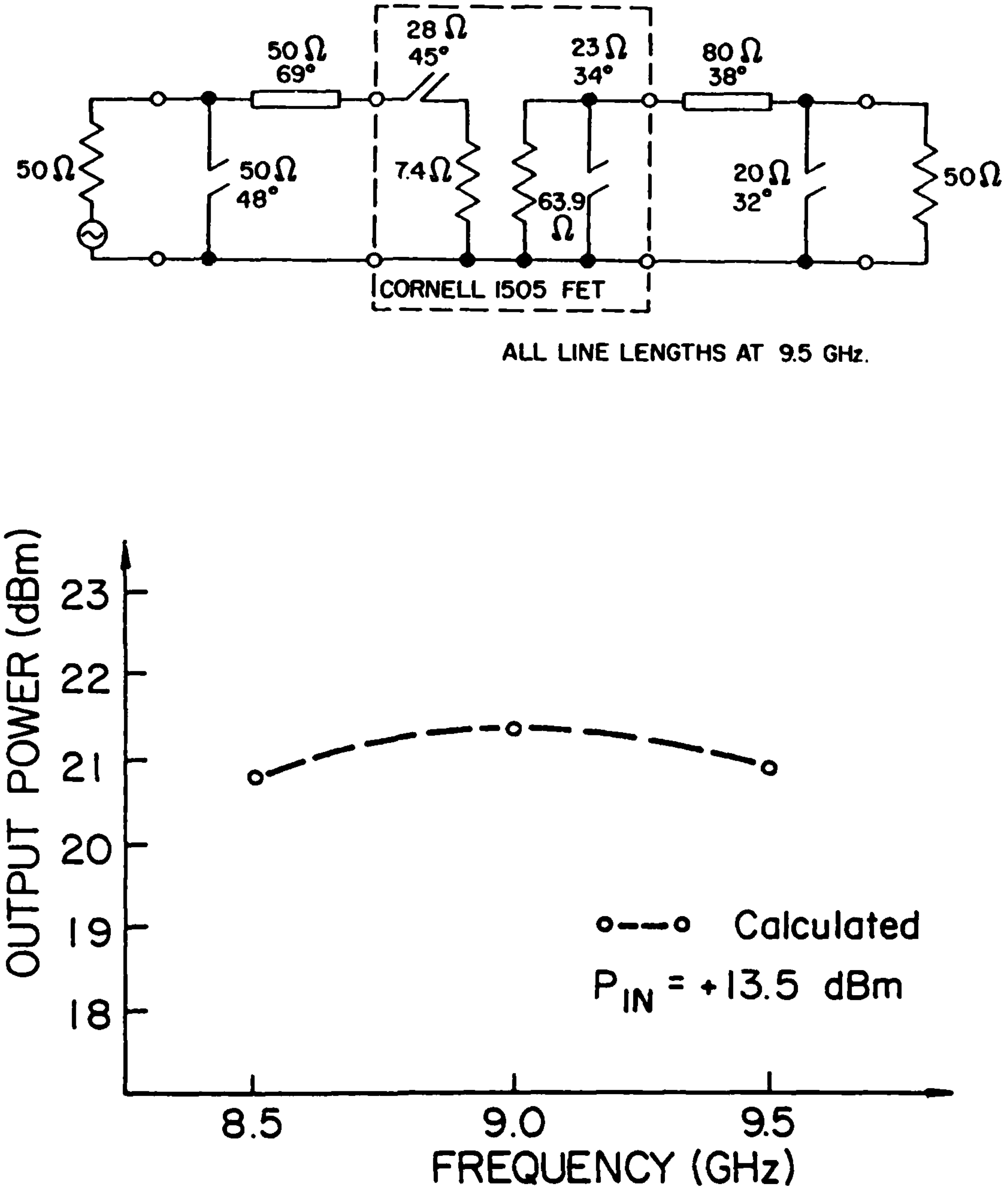
Figure 5.5. Wideband Output Network Performance

Figure 5.6. SEM Photograph of Octave Band Power Amplifier

in the octave band design, above.

The input network shown in Figure 5.7 was formed by producing a single-frequency match at 9.5 GHz, the upper band edge. Analysis showed that this network, too, provided adequate gain slope compensation for the FET. No computer optimization was required for this network.

The output network was formed by producing a single-frequency match at 9 GHz, the band center. This network exhibited 0.1 dB insertion loss at the band edges and 0 insertion loss at 9 GHz; no computer optimization was required.



ALL LINE LENGTHS AT 9.5 GHz.

Figure 5.7. Restricted Topology Monolithic 10% Band Power Amplifier

Figure 5.7 also shows the calculated gain response of this amplifier based on the measured large-signal power contours for the FET. This amplifier achieved a (predicted) 7.5 ± 0.5 dB power gain at its design input drive level of +13.5 dBm. Fabrication of this amplifier was not successful due to processing, so measured results for this design are not available.

Figure 5.8 is a scanning electron microscope photograph of the 8.5 - 9.5 GHz amplifier as fabricated.

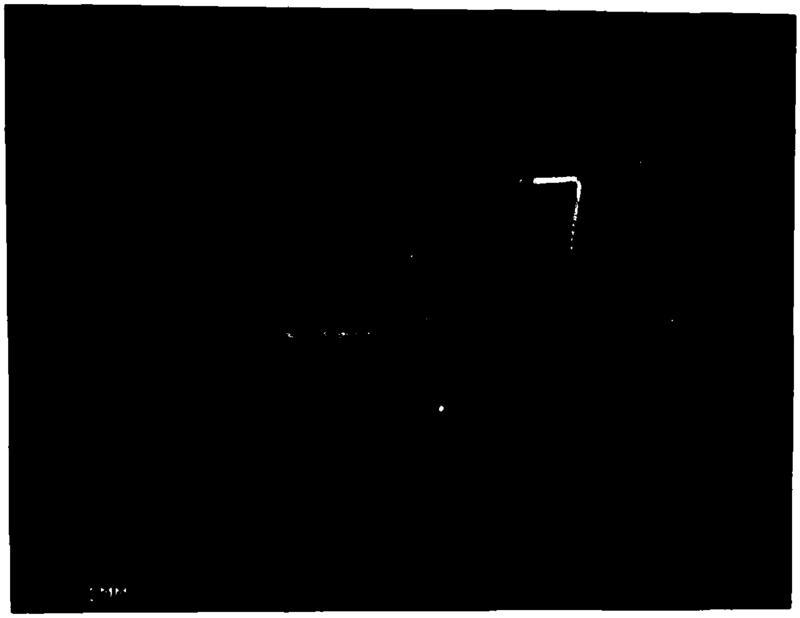


Figure 5.8. SEM Photograph of Monolithic 10% Band Power Amplifier

Chapter 6

Conclusions

The computer-aided procedures for systematic design of lumped element and distributed element amplifier matching networks (limited to even order) have been extended to allow inclusion of odd-order networks. The synthesis method used for these odd-order networks introduces a new synthesis parameter, the denominator polynomial constant (DPC), which provides an added degree of freedom for the circuit designer, making it easier to meet specified circuit performance goals with very simple matching networks (such as 3^{rd} order). The final synthesis algorithm allows specification of either even- or odd-order network topologies for practical computer implementation.

The gain-bandwidth performance of general odd-order networks was discussed, and specific gain-bandwidth results were presented for several different circuit topologies as examples. An algorithm for determining the DPC value which maximizes the gain-bandwidth product of an odd-order network was presented, and practical circuit topologies for monolithic input, output, and interstage networks were discussed along with their design procedures for both lumped and distributed element circuits.

The very significant decrease in monolithic matching network complexity made possible by the use of odd-order networks was demonstrated by examples where 3 element networks were designed to replace much larger 5 element matching networks, usually without significant performance degradation. Both low-noise and power amplifier designs were illustrated. This improvement was made possible by choosing DPC values for the 3rd order networks so that the extra inductor normally required to provide a specific impedance transformation ratio by a Norton Transformation in the network could be eliminated.

An analytical technique was developed for determining the maximum realizable value of the denominator polynomial constant in 3rd order gain functions for both lumped and distributed elements, regardless of ripple, bandwidth, minimum insertion loss, or gain slope specifications. This information is useful to the designer at the time of synthesis, and computer implementations of the procedures were presented that have been incorporated into network synthesis programs.

Chapter 5 presented design techniques and practical topologies for matching networks using only cascaded lines and shunt open-circuited stubs for extremely simple monolithic realizations. A significant limitation with these circuits is the very limited control available over impedance transformation ratio; this problem was resolved by developing a cascaded synthesis technique which includes final computer optimization.

References

[1] W.C. Petersen, "Analytical and Computer-Aided Design Techniques for Bipolar and FET Transistor Amplifiers", PhD Thesis, Cornell University, Ithaca, NY, 1976.

[2] CADSYN User's Manual, COMPACT Engineering, Inc, 1979.

[3] A.N. Riddle and R.J. Trew, "Odd Order Impedance Matching Networks for Low Cost Microwave Integrated Circuits", 1982 IEEE MTT-S International Microwave Symposium Digest, pp. 459-461.

[4] L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, 1962.

[5] G. Dahlquist and A. Bjorck, Numerical Methods, Prentice-Hall, 1974, pp. 220-226.

[6] ibid., pp. 152-157.

[7] J.J. Dongara, C.B. Moler, J.R. Bunch, and G.W. Stewart, LINPACK User's Guide, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[8] E.L. Norton, "Constant-Resistance Networks with Applications to Filter Groups", Bell System Technical Journal, Vol 16, pp. 178-193, 1937.

[9] P.I. Richards, "Resistor-Transmission Line Circuits", Proceedings of the IRE, Vol 36, pp. 217-220, Feb 1948.

[10] H.J. Carlin, "Distributed Circuit Design with Transmission Line Elements", Proceedings of the IEEE, Vol 59, pp. 1059-1081, July 1971.

[11] A. Matsumoto, Microwave Filters and Circuits, Academic Press, 1970.

[12] S. Fu, "Experimental and Computer-Aided Design of Broadband GaAs Monolithic Microwave Power Amplifiers", PhD Thesis, Cornell University, Ithaca, NY, August 1983.

[13] H.W. Bode, Network Analysis and Feedback Amplifier Design, Van Norstrand, New York, 1945.

[14] W. H. Ku and W. C. Petersen, "Optimum Gain-Bandwidth Limitations of Transistor Amplifiers as Reactively Constrained Active Two-Port Networks", IEEE Transactions on Circuits and Systems, Vol CAS-22, No. 6, June 1975.

Appendix A

Source Code Listings

FORTRAN source code listings for the following subroutines are included in this appendix:

A.1 ODDSYN (Lumped element gain function synthesis)

A.2 ODDIST (Distributed element gain function synthesis)

A.3 DLIMTL (Lumped element max DPC calculation)

A.4 DLIMITD (Distributed element max DPC calculation)

A.1 ODDSYN

```
      SUBROUTINE  ODDSYN ( N, XX, RIP, GS, BWR, K, IO, DPC )

C     J. T. DIJAK        16 MAR 1983

      DOUBLE PRECISION  BWR, RIP, GS, X(7), GSP(7), A(7,7), DPC,
     1                  F(7), C(7), XX(22), WO, W1, Z, STEP, XINC, GMAX
      INTEGER   I, J, K, N, IPVT(7), FLAG, P(6,6), PASS, IO
      LOGICAL  ODD, IFLAG

      DATA (P(1,J),J=1,6) / 0,0,0,0,0,0 /
      DATA (P(2,J),J=1,6) / 1,2,0,0,0,0 /
      DATA (P(3,J),J=1,6) / 2,1,3,0,0,0 /
      DATA (P(4,J),J=1,6) / 1,4,2,3,0,0 /
      DATA (P(5,J),J=1,6) / 2,5,1,3,4,0 /
      DATA (P(6,J),J=1,6) / 1,6,3,4,2,5 /

      IF (N .GT. 6)  N = 6
      IF (N .LT. 2)  N = 2

C   SET ODD ORDER FLAG IF N ODD.
      ODD = .TRUE.
      IF (N .EQ. (2*((N+1)/2)))  ODD = .FALSE.
      DO  5  I = 1, N+1
         C(N) = 1.0D0
5     CONTINUE

C   SET UP VECTOR OF CRITICAL FREQUENCIES.
      X(1) = 1.0D0 / (BWR * BWR)
      IF (ODD)  GO TO 15
      X( N+1 ) = 1.0D0
      N2 = N
      XINC = (1.0D0 - X(1)) / DFLOAT(N)
      DO  10  I = 2, N2
         X(I) = X(I-1) + XINC
10    CONTINUE
      GO TO 25

C   ODD- ORDER CRITICAL FREQUENCIES.
15    X(N) = 1.0D0
      N2 = N - 1
      XINC = (1.0D0 - X(1)) / DFLOAT(N-1)
      DO  20  I = 2, N2
         X(I) = X(I-1) + XINC
20     CONTINUE

C   SET (N+1)ST POLYNOMIAL COEFFICIENT IF ODD.
      C(N+1) = DPC

25    IF ( N .LT. 4 )  GO TO 30
         X(2)  = X(2)  - XINC / 2.0D0
         X(N2) = X(N2) + XINC / 2.0D0
```

```
30    N3 = N + 1
      IF ( ODD )  N3 = N
      IF (IO .GT. 0)  WRITE (6.920)  ( X(I), I = 1, N3 )
920   FORMAT ( / ' INITIAL CRITICAL FREQUENCIES :' / 7F10.5 )
      PASS = 1

C   SET UP THE A MATRIX IN SCRAMBLED FORM.
35    DO  40  I = 1, N
         DO  40  J = 1, N
            A(I,J) = X(I) ** P(N,J)
40    CONTINUE

      IF (ODD)  GO TO 65
         DO  45  I = 1, N3
            A(I,N3) = 1.0D0
            A(N3,I) = 1.0D0
45       CONTINUE

C   COMPUTE SPECIFIED GAIN AT CRITICAL FREQUENCIES.
65    IF ( GS .NE. 0.0 )  GO TO 75
         DO  70  I = 1, N3
            GSP(I) = DEXP( -RIP * 0.23025851D0 ) *
     2                           DEXP( RIP * 0.23025851D0 * (-1)**I)
70       CONTINUE
         GO TO 85

75    GSP(N3) = DEXP( -2.0D0 * RIP * 0.23025851D0 )
      N2 = N3 - 1
      DO  80  I = 1,N2
         GSP(I) = GREF( X(I), GS )
         IF ((I.EQ.1).OR.(I.EQ.3).OR.(I.EQ.5))  GSP(I) = GSP(I)*GSP(N3)
80    CONTINUE

C   COMPUTE RIGHT HAND SIDE OF MATRIX EQUATION.
85    Z = 0.0D0
      IF ( ODD )  Z = C(N+1)
      DO  90  I = 1,N3
         IF (K .EQ. 0)  F(I) = 1.0D0 / GSP(I) - Z
         IF (K .NE. 0)  F(I) = ( X(I)**K ) / GSP(I) - Z
90    CONTINUE

C   DO LU FACTORIZATION OF MATRIX A.
      CALL DGEFA ( A, 7, N3, IPVT, FLAG )

      IF (FLAG .EQ. 0)  GO TO 110
         WRITE (6.930)
930      FORMAT ( / ' LU FACTORIZATION FAILURE.' / )
         RETURN

C   SOLVE FOR VECTOR OF COEFFICIENTS.
110   CALL DGESL ( A, 7, N3, IPVT, F, 0 )

C   UNSCRAMBLE POLYNOMIAL COEFFICIENTS INTO DESCENDING ORDER.
```

```
      DO  115   I = 1, N
         C( N+1-P(N,I) ) = F(I)
115   CONTINUE
      IF (.NOT. ODD)   C(N+1) = F(N+1)

C   FIND TRUE CRITICAL FREQUENCIES.
      DO  150  I = 2, N2
C        RESET FLAG SO WE CAN DETERMINE IF ANY INTERATIONS WERE NEEDED.
         IFLAG = .FALSE.
         IF (PASS .LE. 2)  WO = X(I-1)
         IF (PASS .GT. 2)  WO = X(I)
         DO  140  J = 1, 10
120         IF ((I.EQ.2) .OR. (I.EQ.4) .OR. (I.EQ.6))  GO TO 130
125            IF (GPP(WO,C,K,N) .GT. 0.0D0)  GO TO 135
                  WO = 1.02D0 * WO
                  GO TO 125
130            IF (GPP(WO,C,K,N) .LT. 0.0D0)  GO TO 135
                  WO = 1.02D0 * WO
                  GO TO 130
135         STEP = (GREFP(WO,GS) - GP(WO,C,K,N)) /
     2                   (GREFPP(WO,GS) - GPP(WO,C,K,N))
            IF (STEP .LT. -0.03D0)  STEP = -0.03D0
            IF (STEP .GT. +0.03D0)  STEP = +0.03D0
            W1 = WO - STEP
            Z = GREFP( W1,GS ) - GP( W1,C,K,N )
            IF (DABS( Z ) .LT. 0.00005D0)  GO TO 145
            WO = W1
            IFLAG = .TRUE.
140      CONTINUE
145      X(I) = W1
150   CONTINUE

C   HERE ON TERMINATION OF LOOP TO FIND TRUE CRITICAL FREQUENCIES.
      IF (IO .GT. 0)  WRITE (6,940)  (X(I),I = 1, N3)
940   FORMAT ( / ' RE-COMPUTED CRITICAL FREQUENCIES :' / 7F10.5 )

      PASS = PASS + 1
      IF (.NOT. IFLAG)  GO TO 160
      IF (PASS .LT. 10)  GO TO 35

C   JUMP HERE AFTER COMPLETING FINAL PASS THROUGH SYNTHESIS.
160   N2 = N + 1
C        LOAD DENOMINATOR COEFFICIENTS INTO OUTPUT VECTOR.
      DO  165  I = 1, N2
         XX(I) = C(N2 + 1 - I)
165   CONTINUE
C        LOAD CRITICAL FREQUENCIES INTO OUTPUT VECTOR.
      DO  170  I = 1, N3
         XX(N2 + I) = X(I)
170   CONTINUE
      IF (IO .GT. 0 )  WRITE (6,980)  (XX(I), I = 1, N+1 )
980   FORMAT (/ ' DENOMINATOR COEFFICIENTS: (Constant first)' / 7F11.6 )
```

```
      IF (IO .LT. 0)  RETURN
C   FIND MAX VALUE OF GAIN FUNCTION, SO WE CAN NORMALIZE TO 1.0.
      CALL MXGAIN( GMAX,X,C,K,N )

C   NORMALIZE GAIN FUNCTION, AND COMPUTE RESPONSE.
      WRITE (6,950)
950   FORMAT ( / T17, 'GAIN' / T7, 'FREQ', T17, '(DB)' )

      XINC = (1.0D0 - X(1)) / 10.0D0
      DO  180  I = 1, 11
C        IN BAND RESPONSE.
         W1 = X(1) + FLOAT(I-1) * XINC
         G1 = 10.0D0*DLOG10( GAIN2( W1,C,K,N ) / GMAX )
         W2 = DSQRT( W1 )
         WRITE (6,960)  W2, G1
960      FORMAT ( T4, F7.3, F10.3 )
180   CONTINUE
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GAIN2  ( W, C, K, N )
         DOUBLE PRECISION  W, C(7), Z
         INTEGER  N, K
         Z = W * C(1)
         DO  10  I = 2, N
            Z = W * (Z + C(I))
10       CONTINUE
         Z = Z + C(N+1)
         IF (K .EQ. 0)  GAIN2 = 1.0D0 / Z
         IF (K .NE. 0)  GAIN2 = (W**K) / Z
      RETURN
      END

      SUBROUTINE MXGAIN( GMAX, X, C, K, N )
C   FIND MAXIMUM GAIN POINT OVER X(1) TO 2 - X(1) FREQS.
         DOUBLE PRECISION  X(7), C(7), W, GMAX, XINC, G
         INTEGER  N, K
         XINC = (1.0D0 - X(1)) / 40.0D0
         W = X(1)
         GMAX = -25.0D0
         DO  10  I = 1, 80
            W = W + XINC
            G = GAIN2( W, C, K, N )
            IF (G .GT. GMAX )  GMAX = G
10       CONTINUE
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GP ( W, C, K, N )
C        COMPUTE VALUE OF FIRST DERIVATIVE OF GAIN FUNCTION AT W.
         DOUBLE PRECISION  W, C(7), W1, W2
         INTEGER  K, N
         W1 = W + 0.0005D0
         W2 = W - 0.0005D0
         GP = GAIN2(W1,C,K,N) / 0.001D0 - GAIN2(W2,C,K,N) / 0.001D0
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GPP ( W, C, K, N )
C        COMPUTE VALUE OF SECOND DERIVATIVE OF GAIN FUNCTION AT W.
         DOUBLE PRECISION  W, C(7), W1, W2
         INTEGER  K, N
         W1 = W + 0.0005D0
         W2 = W - 0.0005D0
         GPP = GP(W1,C,K,N)/0.001D0 - GP(W2,C,K,N)/0.001D0
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GREF ( W, GS )
C        COMPUTE REFERENCE GAIN AT W.
         DOUBLE PRECISION  W, GS
         GREF = DEXP( 0.38245111D0 * GS * DLOG10( W ) )
      RETURN
      END

      DOUBLE PRECISION FUNCTION GREFP ( W, GS )
C        COMPUTE FIRST DERIVATIVE OF GREF(X) AT W.
         DOUBLE PRECISION W, GS
         GREFP = 0.1660964D0 * GS * GREF( W, GS ) / W
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GREFPP ( W, GS )
C   COMPUTE SECOND DERIVATIVE OF GREF(X) AT W.
         DOUBLE PRECISION  W, GS
         GREFPP = (GS * 0.1660964D0 / W)*( GREFP(W,GS) - GREF(W,GS)/W )
      RETURN
      END
```

A.2 ODDIST

```
      SUBROUTINE  ODDIST ( N, XX, RIP, GS, NPL, K, IO, DPC )

C   THIS ROUTINE IS COMPATIBLE WITH THE DIST PROGRAM AND
C   TAKES THE PLACE OF THE MODNWT ROUTINE.  ODDIST SYNTHESIZES EVEN OR
C   ODD ORDER GAIN FUNCTIONS OF ORDER 2 THROUGH 6 BY ITERATIVELY SOLVING
C   A SYSTEM OF LINEAR EQUATIONS UNTIL BOTH THE GAIN PERFORMANCE AND
C   THE CRITICAL FREQUENCIES OF THE SYNTHESIZED FUNCTION ARE KNOWN.

C   ODDIST DOES FOR DISTRIBUTED CIRCUITS WHAT ODDSYN DOES FOR LUMPED.

C   J. T. DIJAK        16 MAY 1983

      DOUBLE PRECISION  RIP, GS, X(7), GSP(7), A(7,7), DPC,
     1                  F(7), C(7), XX(22), W0, W1, Z, STEP, XINC, GMAX
      INTEGER   I, J, K, N, IPVT(7), FLAG, P(6,6), PASS, IO, NPL
      LOGICAL  ODD, IFLAG

      DATA (P(1,J),J=1,6) / 0,0,0,0,0,0 /
      DATA (P(2,J),J=1,6) / 1,2,0,0,0,0 /
      DATA (P(3,J),J=1,6) / 2,1,3,0,0,0 /
      DATA (P(4,J),J=1,6) / 1,4,2,3,0,0 /
      DATA (P(5,J),J=1,6) / 2,5,1,3,4,0 /
      DATA (P(6,J),J=1,6) / 1,6,3,4,2,5 /

      IF (N .GT. 6)  N = 6
      IF (N .LT. 2)  N = 2

C   SET ODD ORDER FLAG IF N ODD.
      ODD = .TRUE.
      IF (N .EQ. (2*((N+1)/2)))  ODD = .FALSE.
      DO  5  I = 1, N+1
         C(N) = 1.0D0
5     CONTINUE

C   SET UP VECTOR OF CRITICAL FREQUENCIES.
      X(1) = XX(N+2)
      IF (ODD)  GO TO 15
      X( N+1 ) = XX(2*N+2)
      N2 = N
      XINC = (X(N+1) - X(1)) / DFLOAT(N)
      DO  10  I = 2, N2
         X(I) = X(I-1) + XINC
10    CONTINUE
      GO TO 25

C   ODD- ORDER CRITICAL FREQUENCIES.
15    X(N) = XX(2*N+2)
      N2 = N - 1
      XINC = (X(N) - X(1)) / DFLOAT(N-1)
      DO  20  I = 2, N2
         X(I) = X(I-1) + XINC
```

```
20     CONTINUE

C   SET (N+1)ST POLYNOMIAL COEFFICIENT IF ODD.
      C(N+1) = DPC

25    IF ( N .LT. 4 )  GO TO 30
         X(2)  = X(2)  - XINC / 2.0D0
         X(N2) = X(N2) + XINC / 2.0D0

30    N3 = N + 1
      IF ( ODD )  N3 = N
      IF (IO .GT. 0)  WRITE (6,920)  ( X(I), I = 1, N3 )
920   FORMAT ( / ' INITIAL CRITICAL FREQUENCIES :' / 7F10.5 )
      PASS = 1

C   SET UP THE A MATRIX IN SCRAMBLED FORM.
35    DO  40  I = 1, N
         DO  40  J = 1, N
            A(I,J) = X(I) ** P(N,J)
40    CONTINUE

      IF (ODD)  GO TO 65
         DO  45  I = 1, N
            A(I,N3) = 1.0D0
            A(N3,I) = X(N3)**P(N,I)
45       CONTINUE
      A(N3,N3) = 1.0D0

C   COMPUTE SPECIFIED GAIN AT CRITICAL FREQUENCIES.
65    IF ( GS .NE. 0.0 )  GO TO 75
         DO  70  I = 1, N3
            GSP(I) = DEXP( -RIP * 0.23025851D0 ) *
     2                       DEXP( RIP * 0.23025851D0 * (-1)**I)
70       CONTINUE
         GO TO 85

75    Z = DEXP( -2.0D0 * RIP * 0.23025851D0 )
      N2 = N3 - 1
      DO  80  I = 1,N3
         GSP(I) = GREF( X(I), GS )
         IF ((I.EQ.1).OR.(I.EQ.3).OR.(I.EQ.5))  GSP(I) = GSP(I)*Z
80    CONTINUE

C   COMPUTE RIGHT HAND SIDE OF MATRIX EQUATION.
85    Z = 0.0D0
      IF ( ODD )  Z = C(N+1)
      DO  90  I = 1,N3
         IF (K .EQ. 0)  GO TO 88
            IF (NPL .EQ. 0)  F(I) = (X(I)**K)/GSP(I) - Z
            IF (NPL .NE. 0)  F(I)=((X(I)**K)*(X(I)+1.D0)**NPL)/GSP(I)-Z
            GO TO 90
88       IF (NPL .EQ. 0)  F(I) = 1.D0 / GSP(I) - Z
         IF (NPL .NE. 0)  F(I) = (X(I) + 1.D0)**NPL / GSP(I) - Z
```

```
90    CONTINUE

C   DO LU FACTORIZATION OF MATRIX A.
      CALL DGEFA ( A, 7, N3, IPVT, FLAG )

      IF (FLAG .EQ. 0)  GO TO 110
         WRITE (6,930)
930      FORMAT ( / ' LU FACTORIZATION FAILURE.' / )
         RETURN

C   SOLVE FOR VECTOR OF COEFFICIENTS.
110   CALL DGESL ( A, 7, N3, IPVT, F, 0 )

C   UNSCRAMBLE POLYNOMIAL COEFFICIENTS INTO DESCENDING ORDER.
      DO  115   I = 1, N
         C( N+1-P(N,I) ) = F(I)
115   CONTINUE
      IF (.NOT. ODD)   C(N+1) = F(N+1)

C   FIND TRUE CRITICAL FREQUENCIES.
      DO  150   I = 2, N2
C        RESET FLAG SO WE CAN DETERMINE IF ANY ITERATIONS WERE NEEDED.
         IFLAG = .FALSE.
         IF (PASS .LE. 2)  WO = X(I-1)
         IF (PASS .GT. 2)  WO = X(I)
         DO  140  J = 1, 10
120         IF ((I.EQ.2) .OR. (I.EQ.4) .OR. (I.EQ.6))  GO TO 130
125            IF (GPP(WO,C,K,N,NPL) .GT. 0.0D0)  GO TO 135
                  WO = 1.02D0 * WO
                  GO TO 125
130            IF (GPP(WO,C,K,N,NPL) .LT. 0.0D0)  GO TO 135
                  WO = 1.02D0 * WO
                  GO TO 130
135         STEP = (GREFP(WO,GS) - GP(WO,C,K,N,NPL)) /
     2               (GREFPP(WO,GS) - GPP(WO,C,K,N,NPL))
            IF (STEP .LT. -0.03D0)  STEP = -0.03D0
            IF (STEP .GT. +0.03D0)  STEP = +0.03D0
            W1 = WO - STEP
            Z = GREFP( W1,GS ) - GP( W1,C,K,N,NPL )
            IF (DABS( Z ) .LT. 0.00005D0)  GO TO 145
            WO = W1
            IFLAG = .TRUE.
140      CONTINUE
145      X(I) = W1
150   CONTINUE

C   HERE ON TERMINATION OF LOOP TO FIND TRUE CRITICAL FREQUENCIES.
      IF (IO .GT. 0)  WRITE (6,940)  (X(I),I = 1, N3)
940   FORMAT ( / ' RE-COMPUTED CRITICAL FREQUENCIES :' / 7F10.5 )

      PASS = PASS + 1
      IF (.NOT. IFLAG)  GO TO 160
      IF (PASS .LT. 10)  GO TO 35
```

```
C   JUMP HERE AFTER COMPLETING FINAL PASS THROUGH SYNTHESIS.
160   N2 = N + 1
C        LOAD DENOMINATOR COEFFICIENTS INTO OUTPUT VECTOR.
      DO  165  I = 1, N2
         XX(I) = C(N2 + 1 - I)
165   CONTINUE
C        LOAD CRITICAL FREQUENCIES INTO OUTPUT VECTOR.
      DO  170  I = 1, N3
         XX(N2 + I) = X(I)
170   CONTINUE
      IF (IO .GT. 0 )  WRITE (6,980)  (XX(I), I = 1, N+1 )
980   FORMAT (/ ' DENOMINATOR COEFFICIENTS: (Constant first)' / 7F11.6 )

      IF (IO .LT. 0)  RETURN
C   FIND MAX VALUE OF GAIN FUNCTION, SO WE CAN NORMALIZE TO 1.0.
      CALL MXGAIN( GMAX,X,C,K,N,NPL )

C   NORMALIZE GAIN FUNCTION, AND COMPUTE RESPONSE.
      WRITE (6,950)
950   FORMAT ( / T17, 'GAIN' / T7, 'FREQ', T17, '(DB)' )

      XINC = (X(N3) - X(1)) / 10.0D0
      DO  180  I = 1, 11
C        IN BAND RESPONSE.
         W1 = X(1) + FLOAT(I-1) * XINC
         G1 = 10.0D0*DLOG10( GAIN2( W1,C,K,N,NPL ) / GMAX )
         W2 = DSQRT( W1 )
         WRITE (6,960)  W2, G1
960      FORMAT ( T4, F7.3, F10.3 )
180   CONTINUE
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GAIN2  ( W, C, K, N, NPL )
         DOUBLE PRECISION  W, C(7), Z
         INTEGER  N, K, NPL
         Z = W * C(1)
         DO  10  I = 2, N
            Z = W * (Z + C(I))
10       CONTINUE
         Z = Z + C(N+1)
         IF (K .EQ. 0)  GO TO 20
            IF (NPL .EQ. 0) GAIN2 = (W**K) / Z
            IF (NPL .NE. 0) GAIN2 = (W**K)*(W+1.D0)**NPL / Z
            RETURN
20       IF (NPL .EQ. 0)  GAIN2 = 1.D0 / Z
         IF (NPL .NE. 0)  GAIN2 = (1.D0+W)**NPL / Z
      RETURN
      END

      SUBROUTINE MXGAIN( GMAX, X, C, K, N, NPL )
C        FIND MAXIMUM GAIN POINT IN AND ABOVE PASSBAND.
         DOUBLE PRECISION  X(7), C(7), W, GMAX, XINC, G
         INTEGER  N, K, NPL
         XINC = (X(N+1) - X(1)) / 40.0D0
         IF (N .EQ. 3 .OR. N .EQ. 5)  XINC = (X(N)-X(1)) / 40.D0
         W = X(1)
         GMAX = -25.0D0
         DO  10  I = 1, 80
            W = W + XINC
            G = GAIN2( W, C, K, N, NPL )
            IF (G .GT. GMAX )  GMAX = G
10       CONTINUE
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GP ( W, C, K, N, NPL )
C        COMPUTE VALUE OF FIRST DERIVATIVE OF GAIN FUNCTION AT W.
         DOUBLE PRECISION  W, C(7), W1, W2
         INTEGER  K, N, NPL
         W1  = W + 0.0005D0
         W2  = W - 0.0005D0
         GP  = GAIN2(W1,C,K,N,NPL)/0.001D0 - GAIN2(W2,C,K,N,NPL)/0.001D0
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GPP ( W, C, K, N, NPL )
C        COMPUTE VALUE OF SECOND DERIVATIVE OF GAIN FUNCTION AT W.
         DOUBLE PRECISION  W, C(7), W1, W2
         INTEGER  K, N, NPL
         W1  = W + 0.0005D0
         W2  = W - 0.0005D0
         GPP  = GP(W1,C,K,N,NPL)/0.001D0 - GP(W2,C,K,N,NPL)/0.001D0
      RETURN
      END

      DOUBLE PRECISION FUNCTION  GREF ( W, GS )
C        COMPUTE REFERENCE GAIN AT W.
         DOUBLE PRECISION  W, GS
         GREF  = DEXP( 0.38245111D0 * GS * DLOG10( W ) )
      RETURN
      END

      DOUBLE PRECISION FUNCTION GREFP ( W, GS )
C        COMPUTE FIRST DERIVATIVE OF GREF(X) AT W.
         DOUBLE PRECISION W, GS
         GREFP  = 0.1660964D0 * GS * GREF( W, GS ) / W
      RETURN
      END
      DOUBLE PRECISION FUNCTION  GREFPP ( W, GS )
C        COMPUTE SECOND DERIVATIVE OF GREF(X) AT W.
         DOUBLE PRECISION  W, GS
         GREFPP  = (GS * 0.1660964D0 / W)*( GREFP(W,GS) - GREF(W,GS)/W )
      RETURN
      END
```

A.3 DLIMTL

```
      DOUBLE PRECISION FUNCTION  DLIMIT ( RIP, BW, GS, K, X2 )

      DOUBLE PRECISION   RIP, BW, GS, X1, X2, G1, G2, G3, SG22
      X1 = (1.D0 / BW) ** 2

      CALL  SECANT ( X1, RIP, GS, K, X2 )

      G1 = DEXP (0.3824511D0 * GS * DLOG10(X1) - RIP * 0.46051702D0 )
      G2 = DEXP (0.3824511D0 * GS * DLOG10(X2))
      G3 = DEXP ( -RIP * 0.46051702D0 )
      SG22 = 0.1660964D0 * GS / (G2 * X2 )

      T1 = DNUM( X1,X2,G1,G2,G3,SG22,K )
      T2 = DDEN ( X1,X2,K )
      DLIMIT = T1 / T2

      RETURN
      END


      SUBROUTINE SECANT  ( X1, RIP, GS, K, X2 )

      DOUBLE PRECISION  INC, TEMP, B1, B2, B2N, X2L, X2H, X2N,
     2   X1, RIP, GS, X2

      B1 = FUNCT ( X1, X1, RIP, GS, K )
      X2L = X1
      INC = (1.D0 - X1) / 10.D0
      X2H = X2L + INC

C     Search for an argument that produces a sign change from that of B1.
      DO 10  I = 1, 10
         B2 = FUNCT( X1, X2H, RIP, GS, K )

         IF (B2 .EQ. 0.D0)  GO TO 70
         IF (B2 .GT. 0.D0 .AND. B1 .LT. 0.D0)  GO TO 20
         IF (B1 .GT. 0.D0 .AND. B2 .LT. 0.D0)  GO TO 20
         X2L = X2H
         B1 = B2
         X2H = X2H + INC
10    CONTINUE

C     X2L and X2H now bracket the solution. Get improved estimate of solution.
20    TEMP = -1.2D0 * B1 / B2

      IF (TEMP .LT. 1.D0)  X2N = X2L + TEMP * (X2H - X2L) / 2.D0
      IF (TEMP .GT. 1.D0)  X2N = X2H - (X2H - X2L) / (2.D0 * TEMP)

      B2N = FUNCT( X1, X2N, RIP, GS, K )
      IF ( DABS(B2N) .LT. 1.0D-7 )  GO TO 60
      IF (B2N .LT. 0.D0)  GO TO 40
         IF (B1 .GT. 0.D0)  GO TO 30
            B2 = B2N
            X2H = X2N
            GO TO 20
30       B1 = B2N
         X2L = X2N
         GO TO 20

40       IF (B1 .GT. 0.D0)  GO TO 50
            B1 = B2N
            X2L = X2N
            GO TO 20
50       B2 = B2N
         X2H = X2N
         GO TO 20

60    X2 = X2N
      RETURN
70    X2 = X2H
      RETURN
      END

      DOUBLE PRECISION FUNCTION  ZERO  ( C )
C     Zero out each element of array C( ).
      DOUBLE PRECISION  C(6)
      DO  10  I = 1, 6
10       C(I) = 0.D0
      RETURN
      END


      DOUBLE PRECISION FUNCTION  POLY ( C, X )

      DOUBLE PRECISION  C(6), X

C  C(1) is constant coefficient.
C  5th order is maximum assumed.
C  Value of polynomial at X is returned.

      POLY = X * C(6)
      DO  10  I = 5, 2, -1
10       POLY = X * ( POLY + C(I) )
      POLY = POLY + C(1)
      RETURN
      END

      DOUBLE PRECISION FUNCTION  FUNCT ( X1, X2, RIP, GS, K )

      DOUBLE PRECISION  X1,X2,RIP,GS,C(6),G1,G2,G3,SG22,ALPHA,BETA,
     2                     GAMMA, DELTA

      CALL ZERO (C)
      G1 = DEXP( 0.3824511D0 * GS * DLOG10(X1) - RIP * 0.46051702D0 )
      G2 = DEXP( 0.3824511D0 * GS * DLOG10(X2) )
      G3 = DEXP( - RIP * 0.46051702D0 )
      SG22 = 0.1660964D0 * GS / ( G2 * X2 )

      ALPHA = 1.D0/G3
      BETA  = X1**K / G1
      GAMMA = X2**K / G2
      DELTA = K*(X2**(K-1))/G2 - SG22*(X2**K)

      C(1) = BETA - GAMMA - X1*(DELTA - DELTA*X1)
     2          + (GAMMA - ALPHA)*(X1**2)
      C(2) = 2.D0*ALPHA*X1 + DELTA + 2.D0*(GAMMA - BETA)
     2        - (2.D0*GAMMA*X1  + DELTA*(X1**2) )
      C(3) = DELTA * X1 + BETA - DELTA - ALPHA

      FUNCT = POLY ( C, X2 )
      RETURN
      END

      DOUBLE PRECISION FUNCTION  DNUM ( X1,X2,G1,G2,G3,SG22,K )

      DOUBLE PRECISION  X1,X2,G1,G2,G3,SG22,C(6),ALPHA,BETA,GAMMA,DELTA
      CALL ZERO (C)
      ALPHA = 1.D0 / G3
      BETA  = X1**K / G1
      GAMMA = X2**K / G2
      DELTA = K*(X2**(K-1))/G2 - SG22*(X2**K)

      C(1) = GAMMA * (X1**2) * (X1 - 1.D0)
      C(2) = X1*( 2.D0*GAMMA + DELTA*X1 - (DELTA + 2.D0*GAMMA)*(X1**2))
      C(3) = -BETA + X1*( -DELTA - 3.D0*GAMMA + 3.D0*GAMMA*X1 +
     2        (DELTA + ALPHA)*(X1**2) )
      C(4) = 2.D0*BETA + DELTA*X1 - (DELTA + 2.D0*ALPHA)*X1**2
      C(5) = ALPHA * X1 - BETA
      DNUM = POLY ( C, X2 )
      RETURN

      END

      DOUBLE PRECISION FUNCTION  DDEN ( X1, X2, K )
         DOUBLE PRECISION  C(6),X1,X2
         CALL  ZERO (C)
      C(1) = (X1**2) * ( X1 - 1.D0 )
      C(2) = 2.D0 * X1 * ( 1.D0 - (X1**2) )
      C(3) = X1**3 + 3.D0*(X1**2) - 3.D0*X1 - 1.D0
      C(4) = 2.D0 * (1.D0 - X1**2)
      C(5) = X1 - 1.D0
      DDEN = POLY( C, X2 )
      RETURN
      END
```

A.4 DLIMITD

```
      DOUBLE PRECISION FUNCTION  DLIMIT ( RIP, X1, X3, GS, K, Q, X2 )

      DOUBLE PRECISION   RIP, GS, X1, X2, X3, G1, G2, G3, SG22
      INTEGER  Q
      CALL  SECANT ( X1, X3, RIP, GS, K, Q, X2 )

      G1 = DEXP (0.3824511D0 * GS * DLOG10(X1) - RIP * 0.46051702D0 )
      G2 = DEXP (0.3824511D0 * GS * DLOG10(X2))
      G3 = DEXP (0.3824511D0 * GS * DLOG10(X3) - RIP * 0.46051702D0 )
      SG22 = 0.1660964D0 * GS / (G2 * X2 )

      T1 = DNUM( X1,X2,X3,G1,G2,G3,SG22,K,Q )
      T2 = DDEN ( X1,X2,X3 )
      DLIMIT = T1 / T2

      RETURN
      END


      SUBROUTINE SECANT  ( X1, X3, RIP, GS, K, Q, X2 )

      DOUBLE PRECISION  INC, TEMP, B1, B2, B2N, X2L, X2H, X2N,
     2  X1, RIP, GS, X2, X3
      INTEGER  Q

      B1 = FUNCT ( X1, X1, X3, RIP, GS, K, Q )
      X2L = X1
      INC = (X3 - X1) / 10.D0
      X2H = X2L + INC

C     Search for an argument that produces a sign change from that of B1.
      DO 10  I = 1, 10
         B2 = FUNCT( X1, X2H, X3, RIP, GS, K, Q )

         IF (B2 .EQ. 0.D0)  GO TO 70
         IF (B2 .GT. 0.D0 .AND. B1 .LT. 0.D0)  GO TO 20
         IF (B1 .GT. 0.D0 .AND. B2 .LT. 0.D0)  GO TO 20
         X2L = X2H
         B1 = B2
         X2H = X2H + INC
10    CONTINUE

C     X2L and X2H now bracket the solution. Get improved estimate of solution.
20    TEMP = -B1 / B2

      IF (TEMP .LT. 1.D0)  X2N = X2L + TEMP * (X2H - X2L) / 2.D0
      IF (TEMP .GT. 1.D0)  X2N = X2H - (X2H - X2L) / (2.D0 * TEMP)

      B2N = FUNCT( X1, X2N, X3, RIP, GS, K, Q )
      IF ( DABS(B2N) .LT. 1.0D-7 )  GO TO 60
      IF (B2N .LT. 0.D0)  GO TO 40
         IF (B1 .GT. 0.D0)  GO TO 30
            B2 = B2N
            X2H = X2N
            GO TO 20
30       B1 = B2N
         X2L = X2N
         GO TO 20

40       IF (B1 .GT. 0.D0)  GO TO 50
            B1 = B2N
            X2L = X2N
            GO TO 20
50       B2 = B2N
         X2H = X2N
         GO TO 20

60    X2 = X2N
      RETURN
70    X2 = X2H
      RETURN
      END

      DOUBLE PRECISION FUNCTION  ZERO  ( C )
C     Zero out each element of array C( ).
      DOUBLE PRECISION  C(6)
      DO  10  I = 1, 6
10       C(I) = 0.D0
      RETURN
      END


      DOUBLE PRECISION FUNCTION  POLY ( C, X )

      DOUBLE PRECISION  C(6), X

C  C(1) is constant coefficient.
C  5th order is maximum assumed.
C  Value of polynomial at X is returned.

      POLY = X * C(6)
      DO  10  I = 5, 2, -1
10       POLY = X * ( POLY + C(I) )
      POLY = POLY + C(1)
      RETURN
      END

      DOUBLE PRECISION FUNCTION  FUNCT ( X1, X2, X3, RIP, GS, K, Q )

      DOUBLE PRECISION  X1,X2,X3,RIP,GS,C(6),G1,G2,G3,SG22,ALPHA,BETA,
     2                     GAMMA, DELTA
      INTEGER  Q
      CALL ZERO (C)
      G1 = DEXP( 0.3824511D0 * GS * DLOG10(X1) - RIP * 0.46051702D0 )
      G2 = DEXP( 0.3824511D0 * GS * DLOG10(X2) )
      G3 = DEXP( 0.3824511D0 * GS * DLOG10(X3) - RIP * 0.46051702D0 )
      SG22 = 0.1660964D0 * GS / ( G2 * X2 )

      ALPHA = ( X1**K * (1.D0 + X1)**Q ) / G1
      BETA  = ( X2**K * (1.D0 + X2)**Q ) / G2
      GAMMA = ( X3**K * (1.D0 + X3)**Q ) / G3
      DELTA = (K*X2**(K-1)*(1.D0+X2)**Q + Q*X2**K*(1.D0+X2)**(Q-1) )/G2
     2        - SG22 * (X2**K * (1.D0 + X2)**Q )

      C(1) = (X1**2)*(BETA - GAMMA + X3*DELTA) + (X3**2) *
     2             (ALPHA - BETA - DELTA*X1)
      C(2) = 2.D0*(GAMMA - BETA)*X1 + 2.D0*BETA*X3 + DELTA*(X3**2)
     2         - 2.D0*ALPHA*X3  - DELTA*(X1**2)
      C(3) = DELTA * (X1 - X3) + ALPHA - GAMMA

      FUNCT = POLY ( C, X2 )
      RETURN
      END

      DOUBLE PRECISION FUNCTION  DNUM ( X1,X2,X3,G1,G2,G3,SG22,K,Q )

      DOUBLE PRECISION  X1,X2,G1,G2,G3,SG22,C(6),ALPHA,BETA,GAMMA,DELTA,
     2                     X3
      INTEGER  Q

      CALL ZERO (C)
      ALPHA = ( X1**K * (1.D0 + X1)**Q ) / G1
      BETA  = ( X2**K * (1.D0 + X2)**Q ) / G2
      GAMMA = ( X3**K * (1.D0 + X3)**Q ) / G3
      DELTA = (K*X2**(K-1)*(1.D0+X2)**Q + Q*X2**K*(1.D0+X2)**(Q-1) )/G2
     2        - SG22 * (X2**K * (1.D0 + X2)**Q )

      C(1) = (X1**2) * (X3**2) * BETA * (X1 - X3)
      C(2) = X1 * X3 * ( DELTA*X1*(X3**2) - (X1**2)*(DELTA*X3 +
     2       2.D0*BETA) + 2.D0 * BETA * (X3**2) )
      C(3) = (X1**3) * (GAMMA + DELTA*X3) + 3.D0*BETA*(X1**2)*X3
     2        - ALPHA*(X3**3) - X1*(X3**2)*(DELTA*X3 + 3.D0*BETA)
      C(4) = DELTA*X1*(X3**2) + 2.D0*ALPHA*(X3**2)
     2        - (X1**2)*(2.D0*GAMMA + DELTA*X3)
      C(5) = -ALPHA * X3 + GAMMA * X1
      DNUM = POLY ( C, X2 )
      RETURN

      END
```

```
DOUBLE PRECISION FUNCTION  DDEN ( X1, X2, X3 )

DOUBLE PRECISION  C(6),X1,X2,X3

CALL  ZERO (C)

C(1) = (X1**3) * (X3**2) - (X1**2) * (X3**3)
C(2) = 2.D0 * X1 * X3*( X3**2 - X1**2 )
C(3) = X1**3 + 3.D0*(X1**2)*X3 - 3.D0*X1*(X3**2) - X3**3
C(4) = 2.D0 * (X3**2 - X1**2)
C(5) = X1 - X3
DDEN = POLY( C, X2 )
RETURN

END
```

Appendix B

Odd-Order Lumped Element Networks

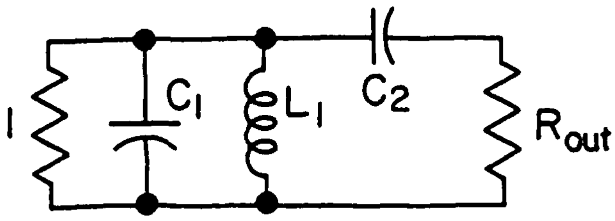
Tables of Element Values

This appendix contains tables of element values for several typical 3^{rd} and 5^{th} order lumped element matching network topologies. Examples are shown for both 15% and 100% (octave) bandwidth.

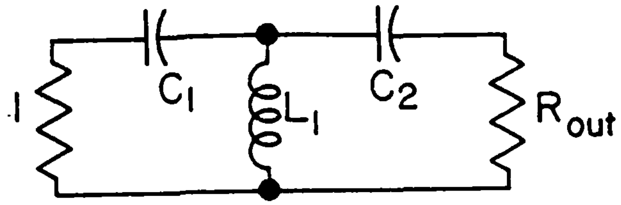
The ripple values listed are in dB, capacitance and inductance values are in Farads and Henries, respectively, and resistances are in Ohms. All values listed are for networks normalized to 1 Ohm source impedance and an upper band edge of 1 Radian per second. To obtain element values for any arbitrary network source impedance, R^* (in Ohms), and any arbitrary upper band edge radian frequency ($\omega^* = 2\pi f^*$), first multiply all resistance and inductance values by R^* and divide all capacitance values by R^*. Then, divide both the capacitance and inductance values by ω^* (Radians/sec).

All networks were synthesized for 0 dB minimum insertion loss, and the denominator polynomial constant was chosen to maximize gain-bandwidth product in all cases.

Attempts were made to synthesize networks with gain slopes (GS) of 0, 3, and 6 dB per octave for each topology. For those topologies where some combinations of ripple and gain slope were unusable, due to gain overshoot or other problems, only those cases which could be synthesized with 0 dB minimum insertion loss are included in the tables.

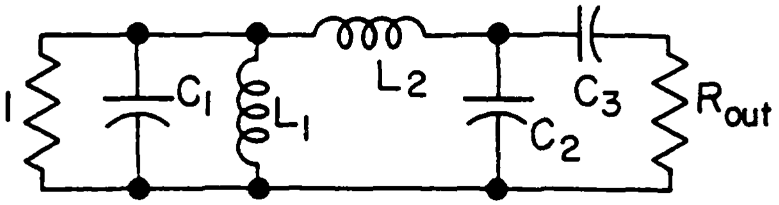


15% Bandwidth				
Rip	C_1	L_1	C_2	R_{out}
GS = 0 dB per octave				
.02	1.961	1.013	0.505	0.622
.05	2.856	0.589	0.516	0.579
.10	3.865	0.396	0.521	0.547
.20	5.321	0.265	0.522	0.534
.50	8.396	0.156	0.524	0.510
1.00	12.358	0.101	0.524	0.496
GS = 3 dB per octave				
.02	2.613	0.816	0.409	0.271
.05	3.019	0.412	0.448	0.666
.10	4.165	0.390	0.470	0.322
.20	5.532	0.264	0.489	0.351
.50	8.538	0.156	0.501	0.367
1.00	12.476	0.102	0.505	0.372
GS = 6 dB per octave				
.02	3.048	0.345	0.428	0.535
.05	3.840	0.328	0.462	0.433
.10	3.988	0.268	0.323	0.853
.20	5.974	0.253	0.428	0.237
.50	8.847	0.154	0.457	0.269
1.00	12.734	0.101	0.466	0.277

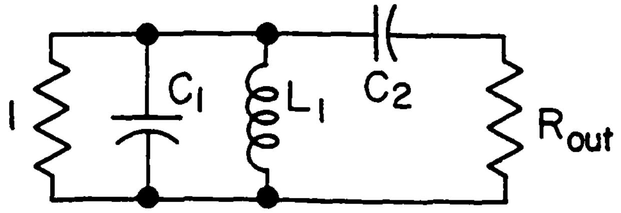


15% Bandwidth

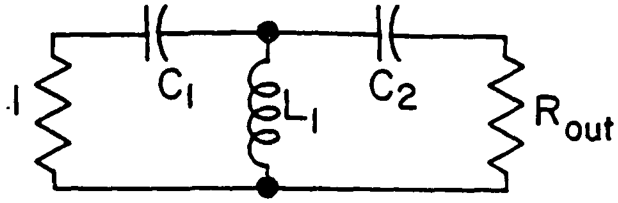
Rip	C_1	L_1	C_2	R_{out}
GS = 0 dB per octave				
.02	0.647	2.033	0.952	3.325
.05	0.446	2.761	0.659	6.304
.10	0.326	3.666	0.464	11.244
.20	0.233	5.051	0.318	21.458
.50	0.144	8.073	0.171	54.995
1.00	0.096	12.033	0.091	122.221
GS = 3 dB per octave				
.05	0.501	2.241	6.718	4.439
.10	0.438	1.963	0.327	3.315
.20	0.241	4.789	6.379	19.527
.50	0.146	7.891	3.886	52.878
1.00	0.097	11.880	2.584	119.777
GS = 6 dB per octave				
.20	0.239	4.561	6.987	18.725
.50	0.146	7.728	4.122	51.940
1.00	0.097	11.744	2.713	118.589



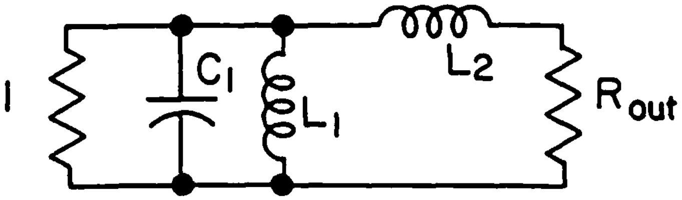
15% Bandwidth						
Rip	C_1	L_1	L_2	C_2	C_3	R_{out}
GS = 0 dB per octave						
.02	5.143	0.226	4.441	0.216	0.108	14.537
.05	6.619	0.175	5.054	0.197	0.073	19.957
.10	8.117	0.143	5.417	0.189	0.056	25.666
.20	10.128	0.114	5.599	0.187	0.044	31.912
.50	14.157	0.082	5.291	0.211	0.071	61.604
1.00	19.309	0.060	4.660	0.242	0.079	75.412
GS = 3 dB per octave						
.02	7.413	0.153	2.737	0.402	0.752	9.083
.05	9.075	0.125	3.278	0.351	3.214	14.272
.10	10.735	0.106	3.730	0.310	66.734	20.343
.20	12.965	0.087	4.084	0.284	68.202	28.475
.50	17.507	0.065	4.175	0.277	75.994	41.861
1.00	23.404	0.048	3.808	0.304	90.357	52.916
GS = 6 dB per octave						
.02	8.635	0.129	2.216	0.527	95.692	6.552
.05	10.420	0.107	2.809	0.416	89.041	11.136
.10	12.189	0.092	3.264	0.358	86.071	16.368
.20	14.552	0.077	3.638	0.321	85.907	23.451
.50	19.369	0.058	3.805	0.307	92.902	35.516
1.00	25.668	0.044	3.518	0.331	108.432	45.742



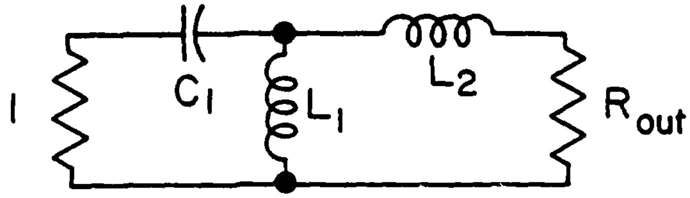
100% Bandwidth				
Rip	C_1	L_1	C_2	R_{out}
GS = 0 dB per octave				
.02	0.355	2.786	5.931	0.915
.05	0.614	2.563	12.516	0.967
.10	0.876	1.991	16.071	0.972
.20	1.255	1.481	21.557	0.976
.50	2.056	0.945	33.753	0.978
1.00	3.090	0.639	49.904	0.979



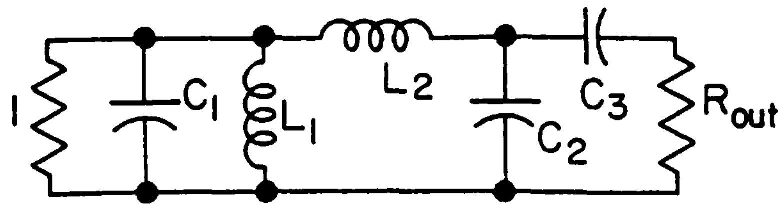
100% Bandwidth				
Rip	C_1	L_1	C_2	R_{out}
GS = 0 dB per octave				
.10	1.445	2.358	1.963	1.864
.20	1.136	2.439	1.158	2.113
.50	0.790	3.149	0.654	3.420
1.00	0.559	4.237	0.346	5.380



100% Bandwidth				
Rip	C_1	L_1	L_2	R_{out}
GS = 0 dB per octave				
.02	0.822	11.876	0.513	0.781
.05	1.094	6.354	0.569	0.707
.10	1.382	3.927	0.601	0.647
.20	1.780	2.507	0.625	0.577
.50	2.595	1.389	0.633	0.466
1.00	3.630	0.890	0.604	0.349
GS = 3 dB per octave				
.02	3.023	2.217	0.354	0.142
.05	3.115	1.772	0.367	0.150
.10	3.271	1.552	0.374	0.150
.20	3.563	1.821	0.356	0.125
.50	4.373	1.087	0.357	0.114
1.00	4.700	0.863	0.364	0.117
GS = 6 dB per octave				
.05	4.653	0.990	0.243	0.067
.10	4.750	0.965	0.248	0.068
.20	4.940	0.917	0.253	0.068
.50	5.510	0.797	0.254	0.063
1.00	6.504	0.640	0.242	0.053

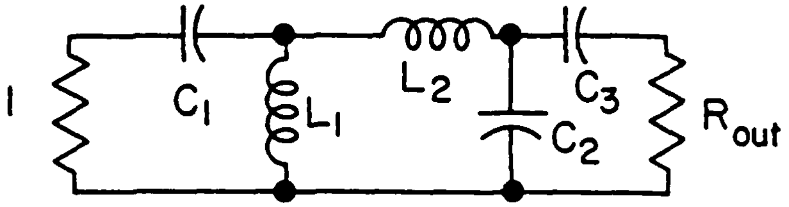


100% Bandwidth				
Rip	C_1	L_1	L_2	R_{out}
GS = 0 dB per octave				
.02	2.452	4.361	0.244	1.371
.05	1.828	3.798	0.328	1.684
.10	1.447	3.839	0.590	2.053
.20	1.124	4.013	1.012	2.698
.50	0.771	4.587	2.060	4.574
1.00	0.551	5.539	3.620	8.169
GS = 3 dB per octave				
.20	0.749	5.060	0.387	1.200
.50	0.629	8.137	2.218	1.708
1.00	0.487	9.007	4.153	2.277
GS = 6 dB per octave				
.50	0.415	11.862	2.947	1.609
1.00	0.356	12.945	4.604	1.903



100% Bandwidth

Rip	C_1	L_1	L_2	C_2	C_3	R_{out}
GS = 0 dB per octave						
.02	1.440	1.633	1.521	0.587	2.147	1.851
.05	1.818	1.253	1.645	0.623	1.592	2.137
.10	2.205	0.991	1.700	0.660	1.274	2.369
.20	2.727	0.785	1.696	0.718	1.039	2.598
.50	3.782	0.555	1.557	0.865	0.828	2.838
1.00	5.136	0.405	1.325	1.091	0.725	2.905
GS = 3 dB per octave						
.05	4.071	0.468	0.733	2.025	62.484	0.591
.10	4.133	0.444	0.775	1.923	53.642	0.693
.20	4.821	0.375	0.832	1.956	32.600	0.883
.50	6.237	0.286	0.830	2.169	93.587	1.169
1.00	8.127	0.217	0.763	2.445	124.443	1.456
GS = 6 dB per octave						
.05	4.816	0.295	0.683	1.802	146.438	0.536
.10	5.441	0.275	0.709	2.044	152.971	0.590
.20	6.266	0.245	0.732	2.244	160.560	0.690
.50	7.968	0.196	0.733	2.498	177.505	0.906
1.00	10.279	0.152	0.672	2.854	206.347	1.119



100% Bandwidth						
Rip	C_1	L_1	L_2	C_2	C_3	R_{out}
GS = 0 dB per octave						
.02	1.389	2.538	2.362	0.195	0.716	3.257
.05	1.100	2.836	3.781	0.167	0.424	4.945
.10	0.907	3.192	5.465	0.139	0.271	7.071
.20	0.733	3.723	8.033	0.113	0.165	10.752
.50	0.529	4.888	13.714	0.082	0.078	21.270
1.00	0.389	6.448	21.101	0.060	0.040	39.015
GS = 3 dB per octave						
.05	0.402	5.948	8.503	0.132	0.428	9.631
.10	0.356	6.640	10.965	0.122	0.403	13.670
.20	0.352	6.375	12.090	0.107	0.285	16.378
.50	0.279	7.813	19.451	0.078	0.179	33.671
1.00	0.217	9.887	29.026	0.057	0.137	65.999
GS = 6 dB per octave						
.05	0.290	6.932	8.390	0.131	0.760	7.915
.10	0.271	7.364	11.320	0.113	0.556	11.571
.20	0.245	8.048	15.597	0.094	0.410	18.070
.50	0.200	9.732	24.584	0.069	0.376	36.382
1.00	0.158	12.183	36.556	0.051	2.092	70.439

Appendix C

Odd-Order Distributed Element Networks

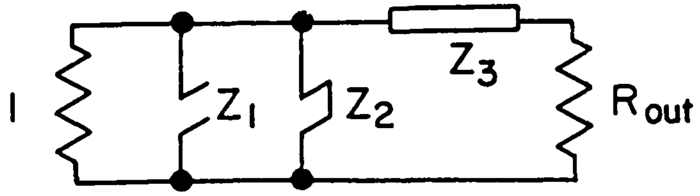
Tables of Element Values

This appendix contains tables of element values for several typical 3rd and 5th order distributed element matching network topologies. Examples are shown for both 15% and 100% (octave) bandwidth.

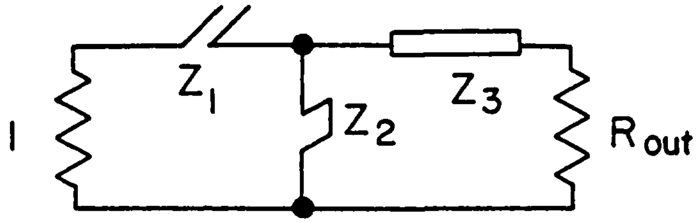
The ripple values listed are in dB; the output impedance and all line impedances are in Ohms, assuming a normalized 1 Ohm network source impedance. To obtain values for some other source impedance, R^* (in Ohms), simply multiply the table entries by R^*. In each case it is assumed that all lines are of the same length, and that this length is electrically one-eighth wavelength at the upper edge of the passband.

All networks were synthesized for 0 dB minimum insertion loss, and the denominator polynomial constant was chosen to maximize gain-bandwidth product in all cases.

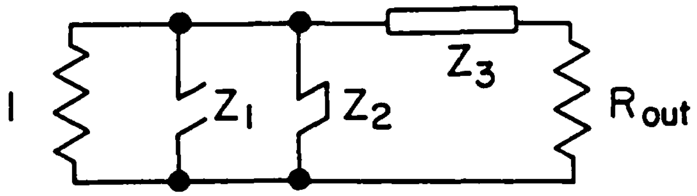
Attempts were made to synthesize networks with gain slopes (GS) of 0, 3, and 6 dB per octave for each topology. For those topologies where some combinations of ripple and gain slope were unusable, due to gain overshoot or other problems, only those cases which could be synthesized with 0 dB minimum insertion loss are included in the tables.



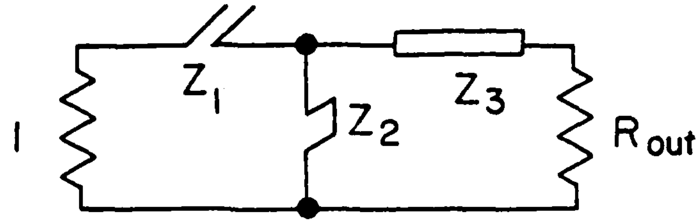
15% Bandwidth				
Rip	Z_1	Z_2	Z_3	R_{out}
GS = 0 dB per octave				
.02	0.825	2.123	0.920	0.545
.05	0.557	1.108	0.892	0.491
.10	0.401	0.704	0.870	0.454
.20	0.285	0.453	0.854	0.431
.50	0.177	0.256	0.831	0.399
1.00	0.119	0.163	0.822	0.387
GS = 3 dB per octave				
.05	0.416	1.169	0.465	0.158
.10	0.287	0.606	0.441	0.169
.20	0.202	0.357	0.435	0.177
.50	0.125	0.189	0.437	0.189
1.00	0.087	0.122	0.441	0.196
GS = 6 dB per octave				
.10	0.247	0.591	0.339	0.093
.20	0.171	0.323	0.325	0.105
.50	0.106	0.165	0.325	0.120
1.00	0.071	0.101	0.328	0.129



15% Bandwidth				
Rip	Z_1	Z_2	Z_3	R_{out}
GS = 0 dB per octave				
.02	1.093	2.451	2.512	2.310
.05	1.562	2.917	4.065	3.350
.10	2.107	3.567	6.471	4.788
.20	2.909	4.553	11.151	7.444
.50	4.626	6.650	25.918	15.774
1.00	6.854	9.405	54.588	31.458
GS = 3 dB per octave				
.10	2.688	5.555	4.764	2.589
.20	3.988	6.927	9.773	5.279
.50	6.568	9.870	25.530	13.689
1.00	9.826	13.783	56.312	29.962

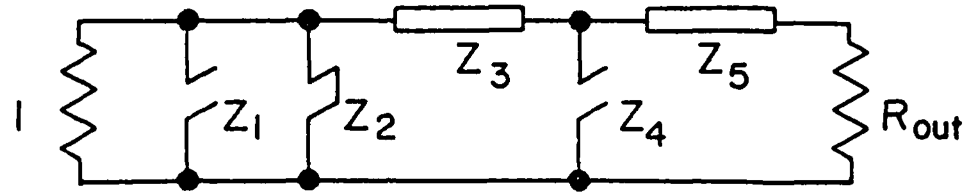


100% Bandwidth				
Rip	Z_1	Z_2	Z_3	R_{out}
GS = 0 dB per octave				
.02	2.097	27.760	1.052	0.774
.05	1.409	13.143	1.049	0.691
.10	1.043	11.950	1.018	0.571
.20	0.772	10.477	0.951	0.439
.50	0.498	2.515	0.930	0.401
1.00	0.345	1.498	0.837	0.298
CS = 3 dB per octave				
.02	0.332	1.042	0.568	0.199
.05	0.317	1.062	0.552	0.182
.10	0.359	1.431	0.519	0.156
.20	0.325	1.961	0.462	0.124
.50	0.261	1.503	0.435	0.103
1.00	0.166	0.888	0.280	0.083
GS = 6 dB per octave				
.02	0.267	0.643	0.414	0.109
.05	0.249	0.591	0.397	0.111
.10	0.241	1.976	0.249	0.049
.20	0.202	1.391	0.228	0.049
.50	0.150	0.827	0.201	0.046
1.00	0.112	0.502	0.180	0.040

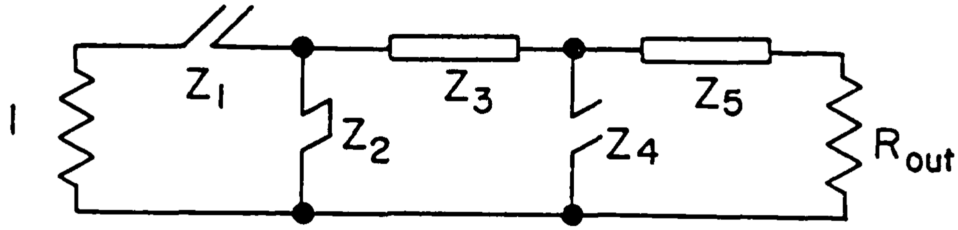


100% Bandwidth

Rip	Z_1	Z_2	Z_3	R_{out}
GS = 0 dB per octave				
.02	0.387	4.636	1.329	1.458
.05	0.471	3.841	1.425	1.693
.10	0.558	3.504	1.562	2.010
.20	0.629	3.584	2.274	2.476
.50	0.908	4.124	3.724	3.957
1.00	1.274	5.050	6.294	6.903
GS = 3 dB per octave				
.50	1.901	12.628	3.295	2.441
1.00	2.760	25.877	6.308	3.070
GS = 6 dB per octave				
.50	2.593	24.433	3.050	1.702
1.00	4.196	22.150	7.514	3.874



100% Bandwidth						
Rip	Z_1	Z_2	Z_3	Z_4	Z_5	R_{out}
GS = 0 dB per octave						
.02	0.851	3.828	1.810	1.052	1.535	1.227
.05	0.669	2.467	1.914	1.032	1.857	1.468
.10	0.549	1.826	1.964	1.008	2.178	1.681
.20	0.442	1.348	1.977	0.992	2.604	2.005
.50	0.317	0.893	1.864	0.925	3.287	2.539
1.00	0.209	0.542	1.447	0.890	2.138	3.584



100% Bandwidth						
Rip	Z_1	Z_2	Z_3	Z_4	Z_5	R_{out}
GS = 0 dB per octave						
.02	0.466	3.301	2.603	3.823	2.695	3.047
.05	0.601	3.379	3.702	3.851	4.157	4.600
.10	0.738	3.554	5.002	4.572	5.994	7.045
.20	1.078	4.503	7.863	7.985	8.162	12.282
.50	1.287	5.063	11.641	6.892	22.308	18.568
1.00	1.736	6.622	17.382	9.151	41.402	23.267
GS = 3 dB per octave						
.05	2.000	9.260	7.104	4.342	5.925	5.674
.10	2.210	9.135	8.949	4.646	8.187	7.157
.20	2.641	10.638	11.704	6.373	11.295	9.391
.50	3.406	11.389	19.920	10.552	24.257	21.427
1.00	4.367	13.557	30.656	15.533	48.925	42.373
GS = 6 dB per octave						
.10	3.375	15.370	9.066	4.089	6.467	5.005
.20	4.125	14.936	14.348	6.496	11.793	9.135
.50	5.352	16.172	25.656	11.516	27.464	21.035
1.00	6.843	18.869	41.074	18.742	58.792	46.087

Appendix D

Single-Frequency Matching Network Design

For narrowband amplifier designs (10% bandwidth or less) it is often possible to obtain satisfactory performance from the impedance matching networks if they are simply designed to achieve the required match at a single frequency. This technique has the advantage of simplicity (the sophisticated computer programs required for broadband synthesis are not needed), and the resulting networks require the fewest possible circuit elements. This technique cannot produce a good match over a wide band, of course, nor can it produce gain responses of arbitrary slope; performance in both of these areas can usually be improved somewhat by computer optimization, however.

D.1 General Design Strategy

The FET matching problem can be viewed as one of matching a capacitive admittance (the input or output of an FET) to a resistive impedance (either the source or load for the amplifier). We wish to use a shunt inductor as the first element connected to the FET (for bias insertion), with the rest of the circuit topology dictated by the desire to use the fewest possible elements to complete the match.

There are two classes of solution to this problem; the magnitude of the real part of the FET admittance controls which of the two solutions may be used in a given case. The methods used to form each of the two types of solutions will be illustrated by examples in the following sections.

D.1.1 C-L Solution Class

When the real part of the normalized FET admittance is less than 1, a simple C-L network may be used to match the device to a resistive source or load. Figure D.1 graphically illustrates the solution method on the Smith Chart.

The FET admittance is represented at A, and we move to point B by adding a shunt inductor. The size of the inductor is chosen so that when we reflect point B into the impedance plane, by moving from B to C, point C falls on the r = 1 circle. Then the match is completed by adding a series capacitor to move from C to D.

Figure D.2 shows the single-frequency C-L networks that result from this procedure which can be used either for input matching (Figure D.2(a)) or output matching (Figure D.2(b)).

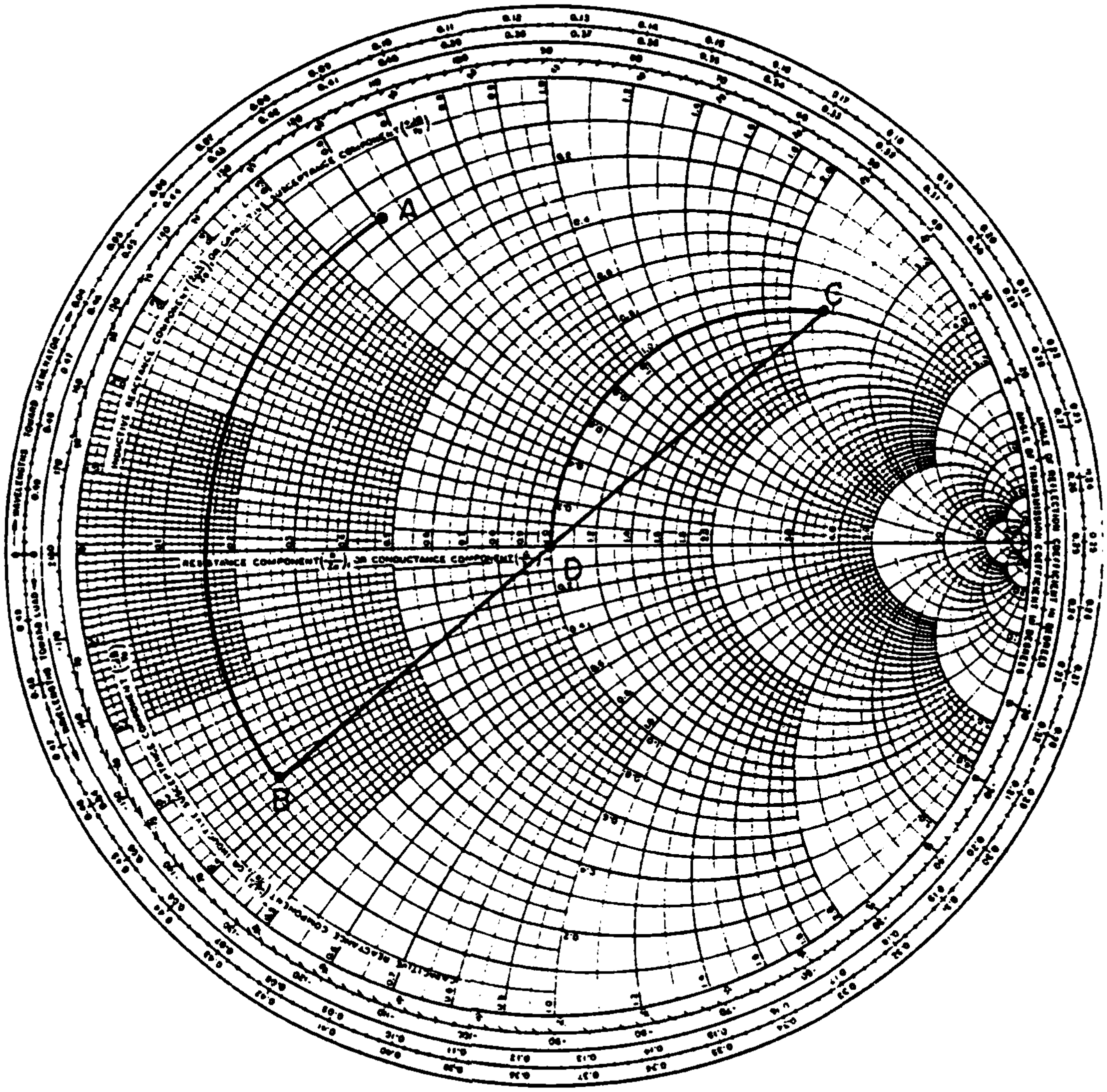


Figure D.1. Smith Chart C-L Network Solution

D.1.2 L-C-L Solution Class

When the real part of the normalized FET admittance is greater than 1, an L-C-L network can be used to accomplish the match. Figure D.3 graphically illustrates the solution method on the Smith Chart.

The FET admittance is again represented at point A, and we move to point B by adding a shunt inductor. This inductor must be sized so that B lies below the zero susceptance axis, but otherwise its value is not critical (this constraint fixes an upper limit on possible values for this

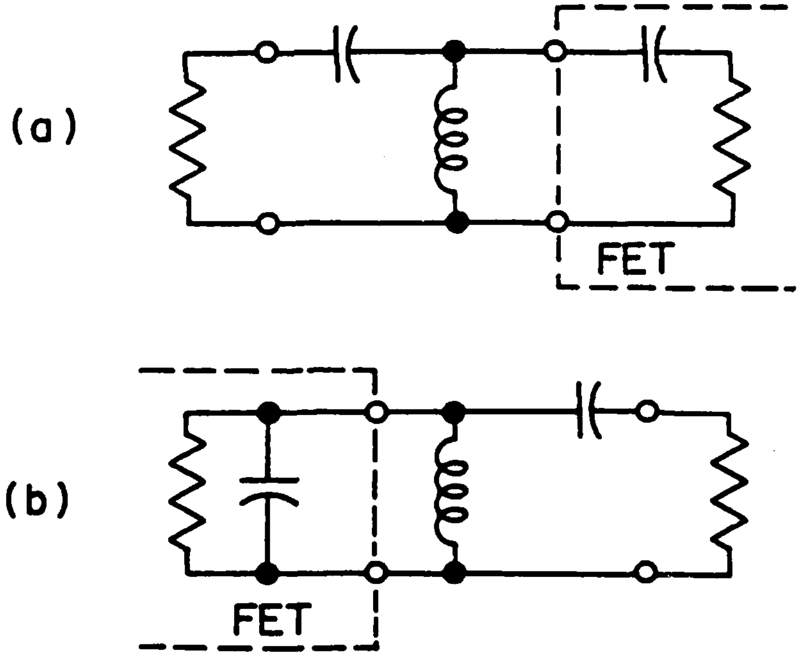


Figure D.2 C-L Input/Output Networks

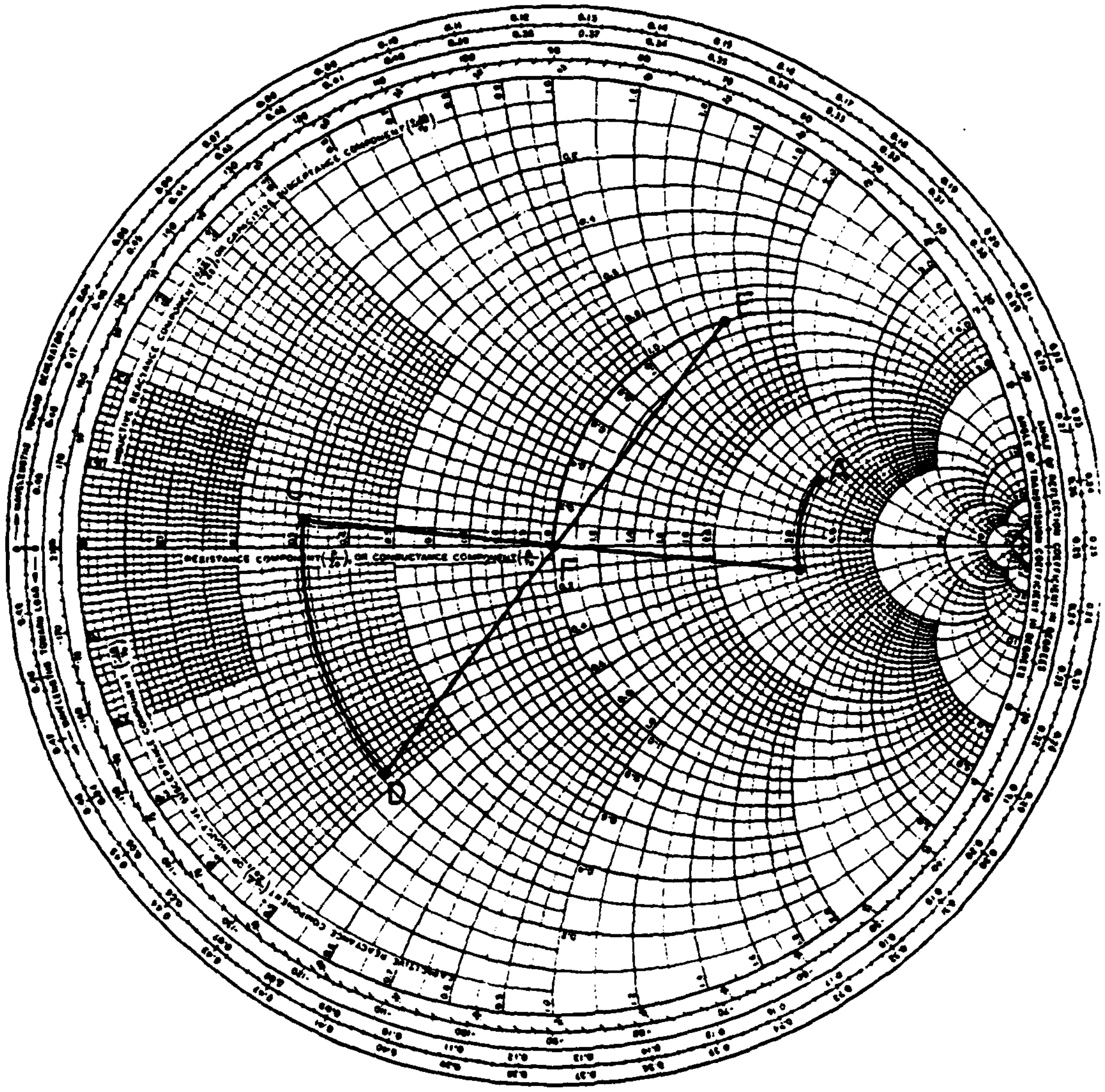

E
A
C
F
D
RESISTANCE COMPONENT ($\frac{R}{Z_o}$), OR CONDUCTANCE COMPONENT ($\frac{G}{Y_o}$)

Figure D.3. Smith Chart L-C-L Network Solution

inductor). We then reflect B to C to move into the impedance plane, and then a series capacitor is added to move from C to D. This capacitor must be sized so that when we reflect point D into the admittance plane by moving from D to E, point E falls on the g = 1 circle. Then the match is completed by adding another shunt inductor to move from E to F.

Figure D.4 shows the single-frequency L-C-L networks that result from this procedure which can be used either for input matching (Figure D.4(a) or output matching (Figure D.4(b)).

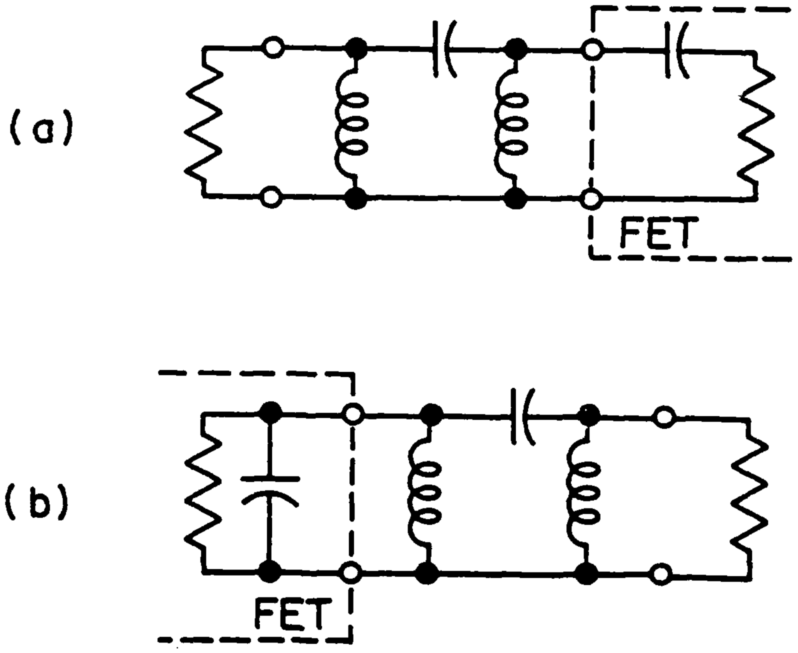


Figure D.4. L-C-L Input/Output Networks

D.2 Interstage Networks

Interstage networks can also be designed by a similar Smith Chart technique. First, a shunt inductor value is determined that transforms the input admittance of the second FET to a pure conductance (or resistance in the impedance plane). Then the output matching problem for the first FET is normalized to this level, and a C-L or L-C-L network, as required, is designed to complete the match to the resistive value now presented at the input to the second FET.

The two circuits are then combined into a single L-C-L network to form the interstage matching network. (If the FET output problem required an L-C-L network, the output inductor of this network is combined with the shunt inductor used initially at the input to the second FET to form a single inductor at that point in the circuit.)

Figure D.5 shows the L-C-L interstage network topology that always results from this procedure.

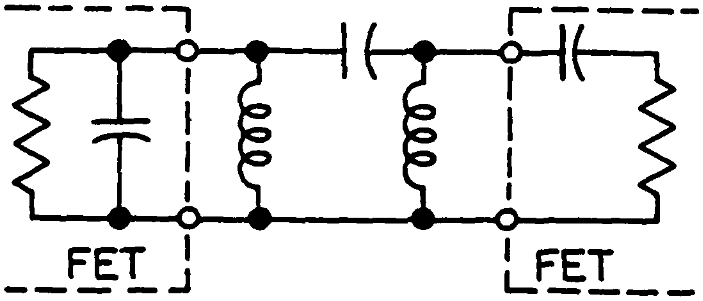


Figure D.5. Single-Frequency Interstage Network

D.3 Conversion to Distributed Inductors

In some circumstances it may be desirable to use shunt short-circuited transmission line stubs in lieu of the lumped inductors used in the above single-frequency solutions. A simple procedure may be used to make this conversion, usually computed at band center. At the chosen frequency (f), compute the inductive reactance (Z_L) of the lumped inductor (L) to be replaced

$$Z_L \;=\; 2\pi f l$$

then choose a line length (θ, in wavelengths) and characteristic impedance (Z_0) for the shorted stub such that

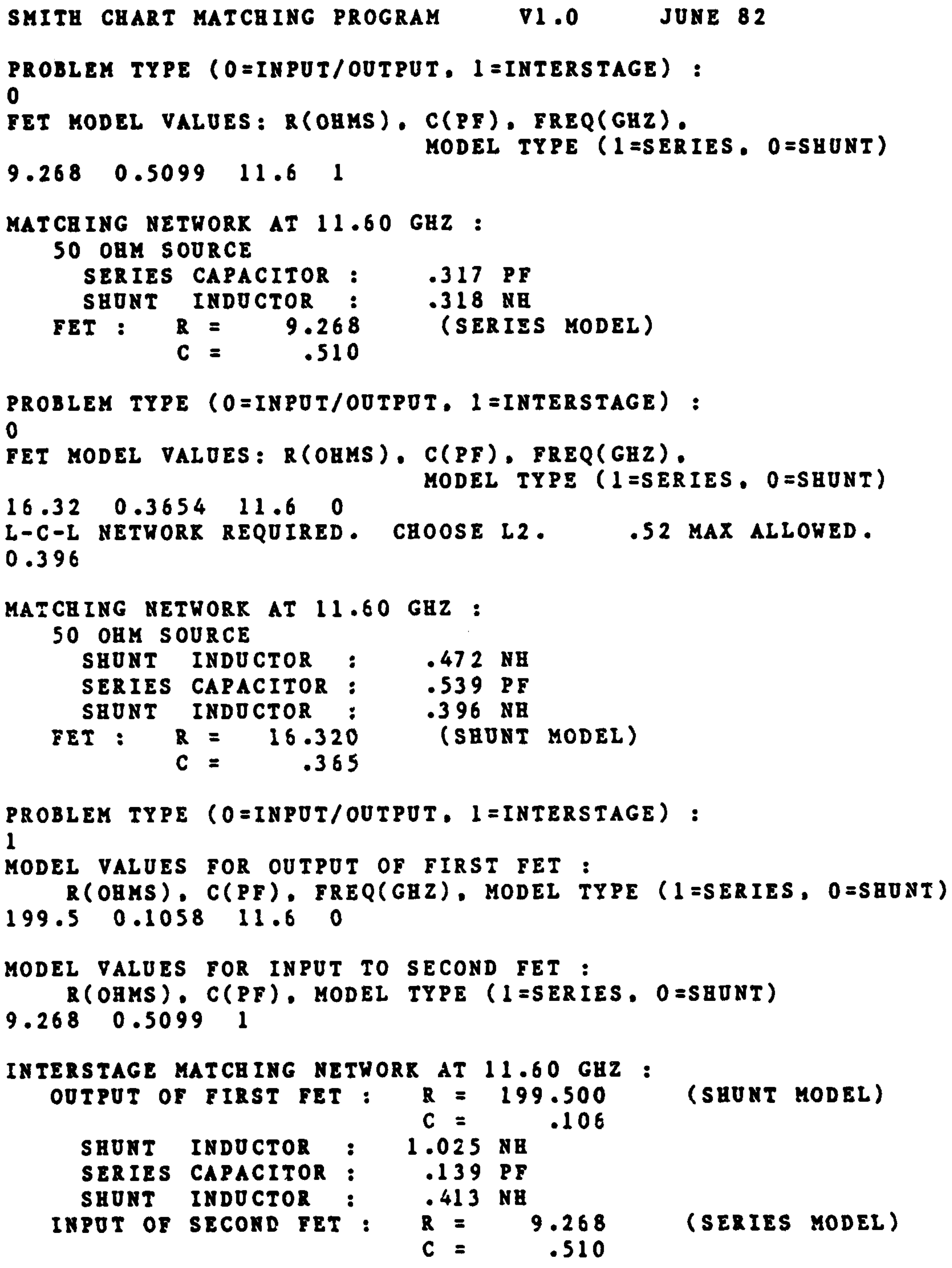
$$Z_L \;=\; Z_0 \tan \theta .$$

Over narrow bands the performance of the circuits after conversion to shunt shorted stubs will be virtually the same as with the lumped inductors; over wider bands, computer optimization of the circuit may be necessary.

D.4 Computer Implementation

Although single-frequency matching problems may be solved graphically, they may also be solved more conveniently by computer. The SMATCH (Smith Chart MATCH) program has been developed to numerically solve the same classes of problems solved graphically above.

The FORTRAN source code listing for the SMATCH program (and its subroutines) follows this section, and Figure D.6 shows example runs of the program for each of the three cases: a C-L network, an L-C-L network, and an interstage network. The program is interactive, as can be seen from the examples, and the solutions are computed and output almost immediately.



```
SMITH CHART MATCHING PROGRAM     V1.0     JUNE 82

PROBLEM TYPE (0=INPUT/OUTPUT, 1=INTERSTAGE) :
0
FET MODEL VALUES: R(OHMS), C(PF), FREQ(GHZ),
                                  MODEL TYPE (1=SERIES, 0=SHUNT)
9.268  0.5099  11.6  1

MATCHING NETWORK AT 11.60 GHZ :
   50 OHM SOURCE
     SERIES CAPACITOR :    .317 PF
     SHUNT  INDUCTOR  :    .318 NH
   FET :   R =    9.268     (SERIES MODEL)
           C =     .510

PROBLEM TYPE (0=INPUT/OUTPUT, 1=INTERSTAGE) :
0
FET MODEL VALUES: R(OHMS), C(PF), FREQ(GHZ),
                                  MODEL TYPE (1=SERIES, 0=SHUNT)
16.32  0.3654  11.6  0
L-C-L NETWORK REQUIRED.  CHOOSE L2.     .52 MAX ALLOWED.
0.396

MATCHING NETWORK AT 11.60 GHZ :
   50 OHM SOURCE
     SHUNT  INDUCTOR  :    .472 NH
     SERIES CAPACITOR :    .539 PF
     SHUNT  INDUCTOR  :    .396 NH
   FET :   R =   16.320     (SHUNT MODEL)
           C =     .365

PROBLEM TYPE (0=INPUT/OUTPUT, 1=INTERSTAGE) :
1
MODEL VALUES FOR OUTPUT OF FIRST FET :
    R(OHMS), C(PF), FREQ(GHZ), MODEL TYPE (1=SERIES, 0=SHUNT)
199.5  0.1058  11.6  0

MODEL VALUES FOR INPUT TO SECOND FET :
    R(OHMS), C(PF), MODEL TYPE (1=SERIES, 0=SHUNT)
9.268  0.5099  1

INTERSTAGE MATCHING NETWORK AT 11.60 GHZ :
   OUTPUT OF FIRST FET :   R =  199.500     (SHUNT MODEL)
                           C =     .106
     SHUNT  INDUCTOR  :  1.025 NH
     SERIES CAPACITOR :    .139 PF
     SHUNT  INDUCTOR  :    .413 NH
   INPUT OF SECOND FET :   R =    9.268     (SERIES MODEL)
                           C =     .510
```

Figure D.6. SMATCH Program Example Runs

```
C   SMITH CHART MATCHING PROGRAM     J.T. DIJAK     21 JUNE 82
C   FILENAME:  SMATCH FORTRAN

      REAL PI,FR,FC,FREQ,FZC,FRN,FZCN,C,L1,L2,L3,YL,FR2,FC2
      COMPLEX YF, ZF
      INTEGER TYPE1, TYPE2

      PI = 3.1415926
      WRITE (6,890)
890   FORMAT (//' SMITH CHART MATCHING PROGRAM',5X,'V1.0',5X,'JUNE 82')

C   CHOOSE PROBLEM TYPE.
5     WRITE (6,900)
900   FORMAT ( / ' PROBLEM TYPE (0=INPUT/OUTPUT, 1=INTERSTAGE) :' )
      READ (5,*,END=1000)  TYPE1
      IF (TYPE1 .EQ. 1)  GO TO 40

C   SOLVE THE INPUT/OUTPUT NETWORK PROBLEM.
      WRITE (6,910)
910   FORMAT ( /' FET MODEL VALUES: R(OHMS), C(PF), FREQ(GHZ), ',
     2   'MODEL TYPE (1=SERIES, 0=SHUNT)' )
      READ (5,*)  FR, FC, FREQ, TYPE1

C   COMPUTE CAPACITIVE REACTANCE OF FET MODEL.
      FZC = 500.0 / (PI * FREQ * FC)
C   NORMALIZE TO 50 OHMS.
      FRN  = FR / 50.0
      FZCN = FZC / 50.0

      IF (TYPE1 .EQ. 1)  GO TO 10
C   SET UP FET SHUNT MODEL.
      YF = CMPLX( 1.0/FRN , 1.0/FZCN )
      GO TO 20

C   SET UP FET SERIES MODEL.
10    ZF = CMPLX( FRN, -FZCN )
      YF = 1.0 / ZF

C   DETERMINE IF C-L OR L-C-L NETWORK IS REQUIRED.
20    IF (REAL(YF) .GT. 1.0)  GO TO 30

C   SOLVE C-L NETWORK FOR ELEMENT VALUES.
      CALL CLNET ( YF, FREQ, 50.0, C, L1 )

C   PRINT SOLUTION.
      WRITE (6,920)  FREQ, C, L1
920   FORMAT ( / ' MATCHING NETWORK AT ', F5.2, ' GHZ :' /
     2   T5, '50 OHM SOURCE' /
     3   T7, 'SERIES CAPACITOR : ', F7.3, ' PF' /
     4   T7, 'SHUNT  INDUCTOR  : ', F7.3, ' NH' )
      IF (TYPE1 .EQ. 1)  WRITE (6,980)  FR, FC
      IF (TYPE1 .EQ. 0)  WRITE (6,982)  FR, FC
980   FORMAT( T5, 'FET :   R = ', F8.3, 5X, '(SERIES MODEL)' /
     2    T13, 'C = ', F8.3 )
982   FORMAT ( T5, 'FET :   R = ', F8.3, 5X, '(SHUNT MODEL)' /
     2    T13, 'C = ', F8.3 )

      GO TO 5

C   SOLVE L-C-L NETWORK FOR ELEMENT VALUES.
30    CALL LCLNET( YF, FREQ, 50.0, L1, C, L2 )

C   PRINT SOLUTION.
      WRITE (6,940)  FREQ, L1, C, L2
940   FORMAT ( /' MATCHING NETWORK AT ', F5.2, ' GHZ :' /
     2   T5, '50 OHM SOURCE' /
     3   T7, 'SHUNT  INDUCTOR  : ', F7.3, ' NH' /
     4   T7, 'SERIES CAPACITOR : ', F7.3, ' PF' /
     5   T7, 'SHUNT  INDUCTOR  : ', F7.3, ' NH' )
      IF (TYPE1 .EQ. 1)  WRITE (6,980)  FR, FC
      IF (TYPE1 .EQ. 0)  WRITE (6,982)  FR, FC

      GO TO 5

C   SOLVE INTERSTAGE PROBLEM.
40    WRITE (6,950)
950   FORMAT (/' MODEL VALUES FOR OUTPUT OF FIRST FET :' /
     2   5X,'R(OHMS), C(PF), FREQ(GHZ), MODEL TYPE (1=SERIES, 0=SHUNT)')
      READ (5,*)  FR, FC, FREQ, TYPE1
      WRITE (6,960)
960   FORMAT (/' MODEL VALUES FOR INPUT TO SECOND FET :' /
     2   5X, 'R(OHMS), C(PF), MODEL TYPE (1=SERIES, 0=SHUNT)' )
      READ (5,*)  FR2, FC2, TYPE2

C   MODEL THE INPUT TO THE SECOND FET AS AN ADMITTANCE YF.
C   COMPUTE CAPACITIVE REACTANCE OF MODEL.
      FZC = 500.0 / (PI * FREQ * FC2)

      IF (TYPE2 .EQ.1)  GO TO 50
C   SHUNT MODEL FOR FET 2.
      YF = CMPLX( 1.0/FR2, 1.0/FZC )
      GO TO 60

C   SERIES MODEL FOR FET 2.
50    ZF = CMPLX( FR2,-FZC )
      YF = 1.0 / ZF

C   RESONATE OUT SUSCEPTANCE OF YF USING SHUNT INDUCTOR L2.
60    YL = AIMAG( YF )
      L2 = 0.5 / (PI * FREQ * YL)

C   COMPUTE NORMALIZATION CONSTANT FOR THE REST OF THE PROBLEM.
      RN = 1.0 / REAL( YF )

C   SET UP NORMALIZED MODEL FOR OUTPUT OF FIRST FET.
      FZC = 500.0 / (PI * FREQ * FC)
```

```
C   NORMALIZE MODEL VALUES.
      FRN  = FR / RN
      FZCN = FZC / RN

      IF (TYPE1 .EQ. 1)  GO TO 70
C   SHUNT MODEL.
      YF = CMPLX( 1.0/FRN, 1.0/FZCN )
      GO TO 80

C   SERIES MODEL.
70    ZF = CMPLX( FRN, -FZCN )
      YF = 1.0 / ZF

C   DETERMINE IF C-L OR L-C-L NETWORK IS REQUIRED.
80    IF (REAL(YF) .GT. 1.0)  GO TO 100

C   SOLVE C-L NETWORK FOR ELEMENT VALUES.
      CALL CLNET( YF, FREQ, RN, C, L1 )

C   PRINT SOLUTION.
90    WRITE (6,970)  FREQ
970   FORMAT ( / ' INTERSTAGE MATCHING NETWORK AT ', F5.2, ' GHZ :' )
      IF (TYPE1 .EQ. 1)  WRITE (6,972)  FR, FC
      IF (TYPE1 .EQ. 0)  WRITE (6,974)  FR, FC
972   FORMAT ( T5, 'OUTPUT OF FIRST FET :   R = ', F8.3, 5X,
     2   '(SERIES MODEL)' / T29, 'C = ', F8.3 )
974   FORMAT ( T5, 'OUTPUT OF FIRST FET :   R = ', F8.3, 5X,
     2   '(SHUNT MODEL)' / T29, 'C = ', F8.3 )
      WRITE (6,976)  L1, C, L2
976   FORMAT ( T7, 'SHUNT  INDUCTOR  : ', F7.3, ' NH' /
     2   T7, 'SERIES CAPACITOR : ', F7.3, ' PF' /
     3   T7, 'SHUNT  INDUCTOR  : ', F7.3, ' NH' )
      IF (TYPE2 .EQ. 1)  WRITE (6,978)  FR2, FC2
      IF (TYPE2 .EQ. 0)  WRITE (6,979)  FR2, FC2
978   FORMAT ( T5, 'INPUT OF SECOND FET :   R = ', F8.3, 5X,
     2   '(SERIES MODEL)' / T29, 'C = ', F8.3 )
979   FORMAT ( T5, 'INPUT OF SECOND FET :   R = ', F8.3, 5X,
     2   '(SHUNT MODEL)' / T29, 'C = ', F8.3 )

      GO TO 5

C   SOLVE L-C-L NETWORK PROBLEM.
100   CALL LCLNET( YF, FREQ, RN, L3, C, L1 )

C   FORM PARALLEL COMBINATION OF L2 & L3.
      L2 = L2 * L3 / (L2 + L3)

C   PRINT SOLUTION.
      GO TO 90

1000  STOP
      END
      SUBROUTINE  SOLVE  ( YF, YL )

C   FIND AN INDUCTIVE SUSCEPTANCE VALUE, YL, SUCH THAT THE
C   NORMALIZED SMITH CHART MATCHING PROBLEM IS SOLVED -
C   FROM A CAPACITIVE ADMITTANCE STARTING POINT, YF.
C   SOLUTION IS GUARANTEED FOR ALL CAPACITIVE YF.
C   THE PROBLEM IS SOLVED BY BISECTION, WITH THE FINAL
C   RELATIVE ERROR IN YL LESS THAN 0.1% .
C      ON ENTRY, YF IS THE CAPACITIVE ADMITTANCE STARTING POINT.
C      ON EXIT , YL IS THE SOLUTION SUSCEPTANCE.

C   J.T. DIJAK      JUNE 1982 //  PART OF SMATCH PROGRAM.

      REAL  YL1, YL2, YL, F, X, Y, T
      COMPLEX YF

C   GENERATE AN INITIAL GUESS FOR YL BASED UPON REAL(YF).  Y IS THE
C   IMAGINARY PART OF THE ADMITTANCE OF THE INITIAL GUESS.
      X = YF
      IF (X .LT. 0.4)  Y = 0.2 + 0.3 * X / 0.4
      IF (X .GE. 0.4 .AND. X .LE. 0.5)  Y = 0.5
      IF (X .GT. 0.5 .AND. X .LE. 0.95) Y = 0.3 + 0.3 * (0.95 - X)/0.45
      IF (X .GT. 0.95) Y = (1.0 - X) * 0.25 / 0.05
C   GENERATE INITIAL GUESS INDUCTIVE SUSCEPTANCE.
      YL1 = Y + AIMAG( YF )

C   WORST CASE ERROR IN THIS INITIAL GUESS IS +/- 0.1.
C   CHOOSE A SECOND STARTING POINT FOR OPPOSITE SIGN OF ERROR FUNCTION.
      F = 1.0 - REAL( 1.0 / (YF - CMPLX(0.0,YL1) ) )
      YL2 = YL1
      IF (F .GT. 0.0)  YL2 = YL2 - 0.15
      IF (F .LE. 0.0)  YL2 = YL2 + 0.15

C   SOLUTION INTERVAL IS NOW DEFINED.  INSURE THAT YL1 CORRESPONDS
C      TO F > 0.
      IF (F .GT. 0.0)  GO TO 10
         T = YL1
         YL1 = YL2
         YL2 = T

C   BISECT THE INTERVAL.
10    YL = ( YL1 + YL2 ) / 2.0

C   EVALUATE THE ERROR FUNCTION.
      F = 1.0 - REAL( 1.0 / ( YF - CMPLX( 0.0, YL ) ) )

C   STOP ONCE THE TERMINATION CRITERIA IS SATISFIED.
      IF ( (YL1 - YL2) .LT. (YL/1000) )  RETURN

C   REPLACE ONE OF THE SOLUTION INTERVAL BOUNDARIES & ITERATE.
      IF ( F .GT. 0.0 )  YL1 = YL
      IF ( F .LT. 0.0 )  YL2 = YL
      GO TO 10
      END

      SUBROUTINE  CLNET ( YF, FREQ, RN, C, L )
C   SOLVE THE C-L NETWORK PROBLEM, WHERE THE NETWORK IS :
C      SERIES C, SHUNT L, FET.
C   ON ENTRY, YF IS STARTING POINT ADMITTANCE, FREQ IS IN GHZ, AND
C      RN IS NORMALIZATION CONSTANT.
C   ON EXIT, C AND L ARE THE NETWORK ELEMENT VALUES IN PF AND NH.

      REAL  C, L, YL, ZC, FREQ, RN
      COMPLEX YF

      PI = 3.1415926
C   SOLVE FOR REQUIRED SHUNT INDUCTIVE ADMITTANCE, YL.
      CALL SOLVE ( YF, YL )
C   COMPUTE SIZE OF SHUNT INDUCTOR (NH).
      L = RN / (2.0 * PI * FREQ * YL)
C   COMPUTE SIZE OF SERIES CAPACITOR (PF).
      ZC = RN * AIMAG( 1.0 / (YF - CMPLX( 0.0, YL ) ) )
      C  = 500.0 / (PI * FREQ * ZC)
      RETURN
      END

      SUBROUTINE  LCLNET ( YF, FREQ, RN, L1, C, L2 )
C   SOLVE THE L-C-L NETWORK PROBLEM, WHERE THE NETWORK IS :
C      SHUNT L1, SERIES C, SHUNT L2, FET.
C   ON ENTRY, YF IS STARTING POINT ADMITTANCE, FREQ IS IN GHZ, AND
C      RN IS NORMALIZATION CONSTANT.
C   ON EXIT, L1, C, AND L2 ARE THE NETWORK ELEMENT VALUES IN PF & NH.

      REAL  FREQ, RN, L1, C, L2, ZC, L2MAX, YL1, PI
      COMPLEX YF, Y1, Z1

      PI = 3.1415926
C   FIND MAX ALLOWABLE VALUE FOR THE SHUNT INDUCTOR L2 (NEXT TO FET).
C   THIS VALUE FOR L2 WILL MOVE YF JUST TO THE REAL AXIS.
      L2MAX = RN / (2.0 * PI * FREQ * AIMAG(YF) )
      WRITE (6,900)  L2MAX
900   FORMAT ( / ' L-C-L NETWORK REQUIRED.  CHOOSE L2. ', F7.2,
     2   ' MAX ALLOWED.' )
      READ (5,*)  L2
      IF (L2 .GT. L2MAX)  L2 = 0.95 * L2MAX

C   COMPUTE RESULTING ADMITTANCE AFTER ADDING SHUNT L2.
      Y1 = YF - CMPLX( 0.0, (RN/(2.0 * PI * FREQ * L2) ) )
      Z1 = 1.0 / Y1
C   SOLVE THE MATCHING PROBLEM IN THE Z-PLANE RATHER THAN Y-PLANE.
      CALL SOLVE( Z1, ZC )
```

```
C   COMPUTE SIZE OF SERIES CAPACITOR (PF).
      C = 500.0 / (PI * FREQ * ZC * RN)
C   COMPUTE SIZE OF SECOND SHUNT INDUCTOR, L1 (NH).
      YL1 = AIMAG( 1.0 / (Z1 - CMPLX( 0.0,ZC ) ) )
      L1 = RN / (2.0 * PI * FREQ * YL1)

      RETURN
      END
```